DM 7,- ÖS 62,- Sfr. 7,- Oktober '87 Nr. 10

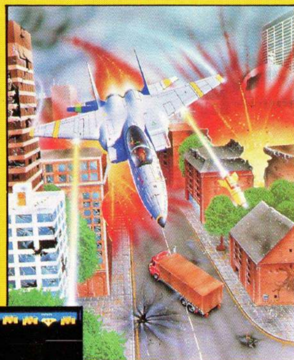
AMIGA

KICK START



BILLIG-SPIELE FÜR AMIGA und ST MÜSSEN NICHT »BILLIG« MÜLL SEIN!

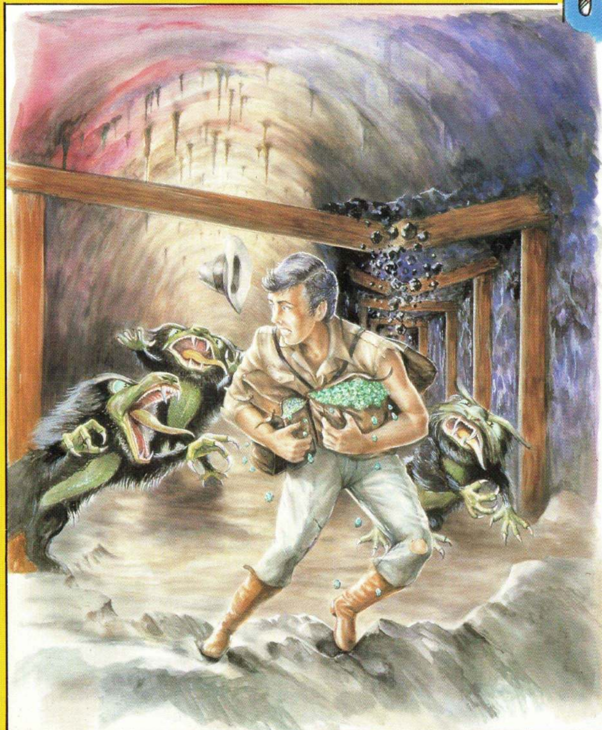


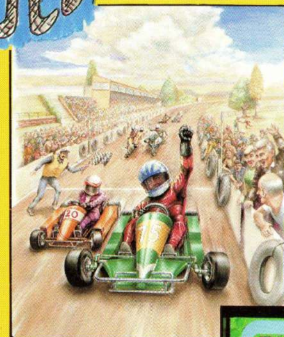


CITY DEFENCE

von Uwe Kühner

Eine schier endlose Zahl tödlicher Laserstrahlen aus dem Weltall kann nur gestoppt werden, wenn Sie zum richtigen Zeitpunkt Ihre Bodenraketen abschiessen, bevor die Städte zerstört werden. 1/2 Spieler; Maussteuerung. **Lieferbar für AMIGA**



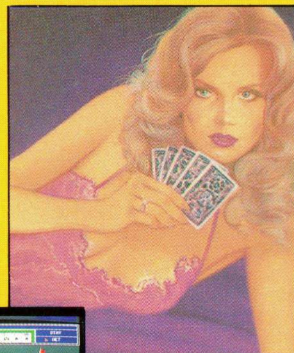


GOKART RACING

von Anco

Ein exzellentes Autorennen, bei dem ein echter Champion gefordert wird. Bei unterschiedlichsten Witterungsbedingungen und Strecken müssen Sie vorher Reifen und Übersetzung wählen. 8 Kurse, 1/2 Spieler (gleichzeitig); Joysticksteuerung.

Lieferbar für AMIGA und ST



STRIP POKER

von Artworx

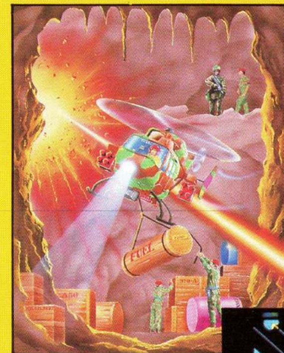
Verbringen Sie einen schönen Abend mit einer Runde Strip Poker. Mit Susi und Melissa stehen Ihnen 2 bildschöne, aber spielstarke Partnerinnen zur Verfügung. Bequeme Spielsteuerung mit der Maus. **Lieferbar für AMIGA und ST**

EMERALD MINE

von Klaus Heinz und Volker Wertich

Ein Riesen-Arcade-Adventure mit über 100 Leveln und 2-Spieler-Option (gleichzeitig im Teamwork!); Joysticksteuerung. *"Die flotte Grafik mit witzigen Monster-Sprites und vor allem der Sound gefallen mir sehr gut: Die fetzige Titelmusik und die digitalisierten Sound-Effekte sind exzellent. Das Verblüffendste an diesem tollen Spiel ist aber der sehr niedrige Preis: Wer hier nicht zuschlägt, ist selber schuld." (HAPPY COMPUTER)*

Lieferbar für AMIGA und ST

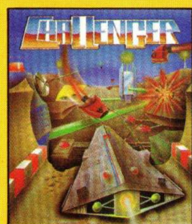


FORTRESS UNDERGROUND

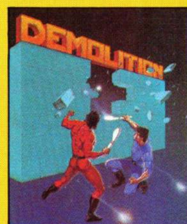
von Andreas von Lepel

In einer gigantischen unterirdischen Höhle (640 Bildschirme groß) müssen Sie ein feindliches Kraftwerk aufspüren und eliminieren. Aber der Weg dorthin ist lang und gefährlich. Joysticksteuerung.

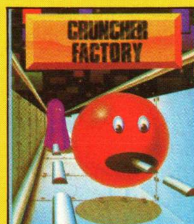
Lieferbar für AMIGA



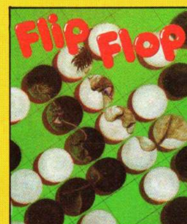
Lieferbar für AMIGA



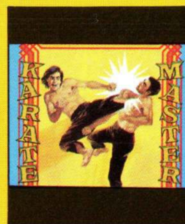
Lieferbar für AMIGA



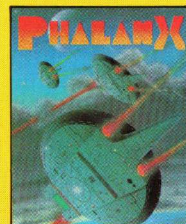
Lieferbar für AMIGA



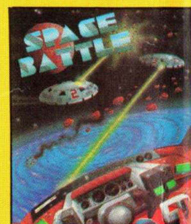
Lieferbar für AMIGA/ST



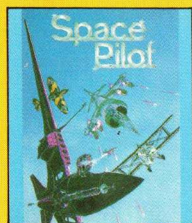
Lieferbar für ST



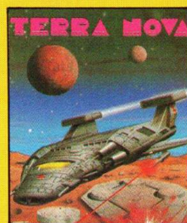
Lieferbar für AMIGA



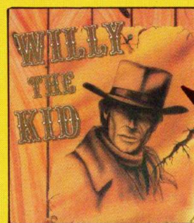
Lieferbar für AMIGA



Lieferbar für ST



Lieferbar für ST



Lieferbar für AMIGA/ST

KINGSOFT-Spiele gibt's überall da, wo es gute Software gibt.



SPITZEN-SOFTWARE ...natürlich von

KINGSOFT

MADE IN GERMANY



Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA und ST - am besten kostenlos den aktuellen Katalog anfordern!

Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung, Super-Grafik und fantastischen Digisound-Effekten.

F. Schäfer · Schnackebusch 4 · 5106 Roetgen
☎ 02408/5119 (nicht aufgeben!) Fax 02408-5213



EDITORIAL

Liebe Leser, liebe Leserinnen!

Was macht die Software?

Die Frage nach Software auf dem Amiga wurde oft gestellt. Besonders im Anfangsstadium des Rechners konnte man die Frage nicht ohne weiters beantworten. Einige Softwarehäuser entwickelten von Anfang an fleißig Programme, andere wiederum wußten nicht so recht wo der Amiga einzuordnen war, bzw. ob er sich im Markt etabliert und ließen deshalb zunächst die Finger vom 'heißen Eisen' Amiga. Durch die Produkterweiterung des Amiga um zwei weitere Modelle (500 und 2000) wurde jegliche Skepsis beseitigt und renomierte sowie neue Softwareproduzenten begannen Programme zu entwickeln.

Einige Firmen hatten nur den Gedanken, so schnell wie möglich auf den Amiga-Zug aufzuspringen. Dementsprechend war die Qualität der Produkte, vom Preis ganz zu schweigen. Vorsicht scheint geboten! Zu schnell bekommt man schlechte Produkte untergeschoben.

Wir, die KICKSTART Redaktion, bemühen uns deshalb immer objektive und informative Tests zu schreiben, sodaß Sie, liebe Leser und Leserinnen, genau über die Qualität und den Nutzen der verschiedenen Produkte (nicht nur auf Software, sondern auch auf Hardware bezogen) informiert werden.

Speziell im Spiele-Bereich geht es recht munter zu. Eine wahre Flut von Produkten scheint im Anmarsch. Man darf auf die Qualität gespannt sein. Aber auch die professionelle Anwendung findet inzwischen qualitative Software, besonders im Grafik- und Animations-Bereich.

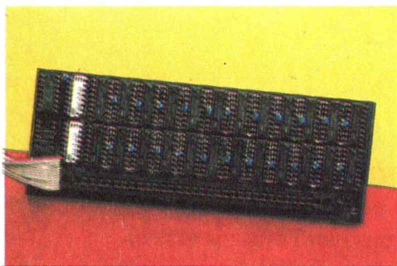
Trotz der zur Zeit schon erhältlichen sehr guten Software glaube ich, daß die Qualität in der nächsten Zeit noch gewaltig steigen wird. Zu wenig Programmierer können den Amiga mit den herausragenden Hardwareeigenschaften bis ins letzte Bit programmieren. Aber dieser Mangel wird hoffentlich bald beseitigt sein.

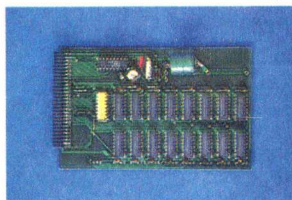
Andreas Krämer

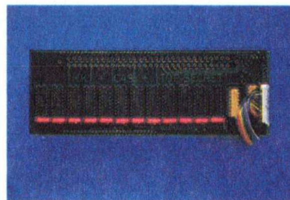
A. Krämer

Inhalt:

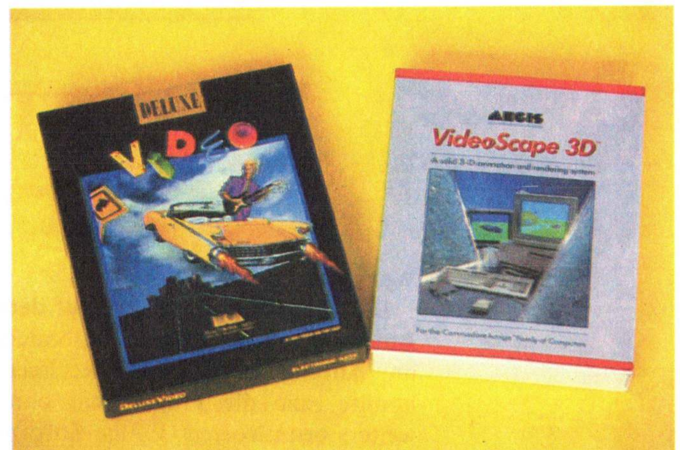
SOFTWARE







HARDWARE



MEGA AMIGA
Durch die aufwendige und speicherintensive Grafikverwaltung des AMIGA reicht ein Speicher von 512 K meistens nicht mehr aus. Es ist daher angebracht, einen Ausbau auf 1 MByte durchzuführen. Wir stellen zwei Erweiterungen für den AMIGA 1000 und auch eine für den AMIGA 500 vor.

ANIMATION kontra MANIPULATION
VIDEO SCAPE 3D von AEGIS und DELUXE VIDEO V1.2 von ECA zählen sicherlich zu den interessantesten Produkten für den AMIGA. Aber auch der BUTCHER hat in Bezug auf Grafikverarbeitung einiges zu bieten.



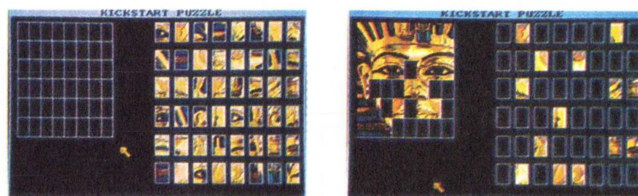
MS-DOS

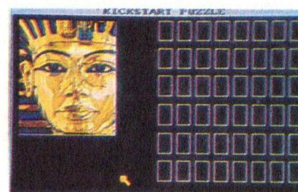
GRUNDLAGEN

SPIELE

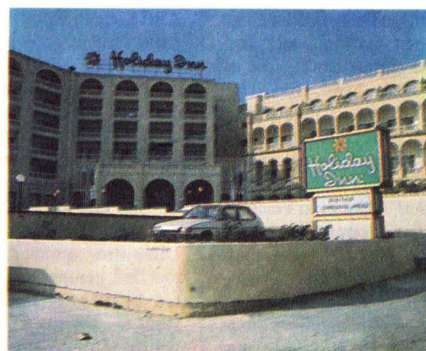


LISTING





Sag mir wo die Teile sind: Kickstart PUZZLE
Ihr Lieblingsbild zerlegt in 1000 Teile, welche Vorstellung. Aber sicherlich sind Sie in der Lage – mit viel Geduld – das Bild wieder zusammenzusetzen.



RUBRIKEN:



QUIZAM

Quizam ist eine amerikanische Version von Trivial Pursuit. Teilnehmen können bis zu acht Spieler. Das Besondere an Quizam ist die Möglichkeit, eigene Fragen und Sachgebiete einzugeben und abzuspeichern. Die Graphik ist recht farbenprächtig. Dagegen treibt die sich ständig wiederholende Musiksequenz auch den härtesten Sünder in den Beichtstuhl.

Bezugsadresse:
Interstel Corp.
PO.BOX 57825
Webster TX 77598
USA

TYPHOON

Ein kleines Raumschiff fliegt von unten nach oben. Ihm entgegen fliegen feindliche Raumschiffe, die es abzuschießen gilt. Nein, Sie irren sich. Das ist nicht... Dieses Spiel heißt Typhoon, ist recht langsam und unterbricht oft mitten in einem Level, um nachzuladen. Außerdem läßt es sich nur laden, wenn man Peripheriegeräte wie Zweitfloppy abmontiert. Mir persönlich hat die vorliegende Version noch nicht gefallen – vielleicht wird sie ja noch verbessert. Mit einigen Änderungen könnte KINGSOFT ein Spiel daraus machen, das bei Ballerspiel-Fans sehr gut ankommt.

RAINBOW-ARTS-WETTBEWERB

In der letzten Ausgabe dieser Zeitschrift haben wir eine Besprechung von BAD CAT veröffentlicht. BAD CAT ist ein neues Spiel der Softwarefirma Rainbow Arts. Der Titelheld der Geschichte will an der Katzenolympiade in Los Angeles teilnehmen. Sein Markenzeichen ist ein... Stirnband.

Wenn Sie die Farbe des Stirnbandes wissen, schreiben Sie sie doch ganz einfach auf eine Postkarte und schicken diese bis zum 15. Oktober (Datum des Poststempels) an die Redaktion der Kickstart. Unter den Einsendern verlosen wir 20 Originale des olympischen Abenteuers unseres Helden BAD CAT.

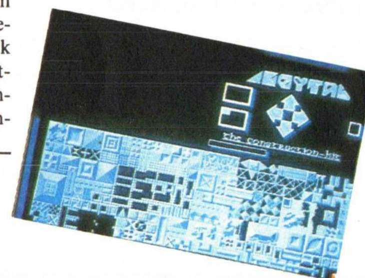
Für all diejenigen, die noch eine Diskette mit einem kleinen selbstgeschriebenen Public-Domain-Programm mitschicken, stellt RAINBOW ARTS einen Sonderpreis zur Verfügung. Wir verlosen unter ihnen einen einwöchigen Besuch bei RAINBOW ARTS mit Informationen, Programmiertips, Spielen etc. Auf der eingesandten Diskette erwarten wir GANZ EINFACHE Spielchen, Utilities und ähnliches, die frei von Rechten Dritter sind und als Public-Domain-Software weitergegeben werden können. Diese Programme dürfen nicht kommerziell genutzt werden. Wir erwarten keine programmierten Superleistungen. Da ein einwöchiger Besuch bei RAINBOW ARTS aber gleichzeitig eine Art „Bildungsurlaub" in Sachen Programmieren darstellt, soll er an jemanden fallen, der wenigstens weiß, wie man Pasick, Tse oder Ahsemmpler schreibt. Auch für den Sonderpreis gilt der Einsendeschluß 15. Oktober 1987 (Datum des Poststempels). Bei beiden Verlosungen sind Mitarbeiter der Redaktion ausgeschlossen. Ausgeschlossen ist auch der Rechtsweg. Wir bitten um Verständnis dafür, daß wir für die Rücksendung eingeschickter Disketten keine Garantie übernehmen können.

DISCOVERY SOFTWARE

Zwei neue Spiele von Discovery werden bis Mitte September fertig sein: ULTRABALL und AEGITAS. Die Ultraball zugrundeliegende Idee könnte ursprünglich ähnlich der von „Break" oder „Arkanoid" gewesen sein. Inzwischen ist das Spiel aber weit über diese „Vorbilder" hinausgewachsen. Sound, Grafik und Action sind wirklich sehr gut. Das zweite Spiel, AEGITAS, ist ein Actionspiel der Kategorie Schlacht im Weltraum. Zum Programmieren dieses Spiels wurde eine neue Technik angewandt – die Levels werden mittlerweile nicht mehr komplett gezeichnet, sondern mit Hilfe eines „Construction Sets" designed. Die feindlichen Raumflotten werden ähnlich erstellt. Jedes Level hat ein Ausmaß von 3x3 Bildschirmen. Bei Redaktionsschluß war der Sound noch nicht in das Spiel eingebunden, so daß wir ihn nicht beurteilen konnten. Grafik und Animation sind sehr gut. In Arbeit ist zur Zeit auch ein weiteres Projekt. Sein Arbeitstitel: SOCCER ELITE. Alle Fußballfans werden diese Nachricht sicher gerne hören.

Bezugsadresse:
Discovery Software
Vertrieb Deutschland
Crevennastr. 11 a
8700 Würzburg



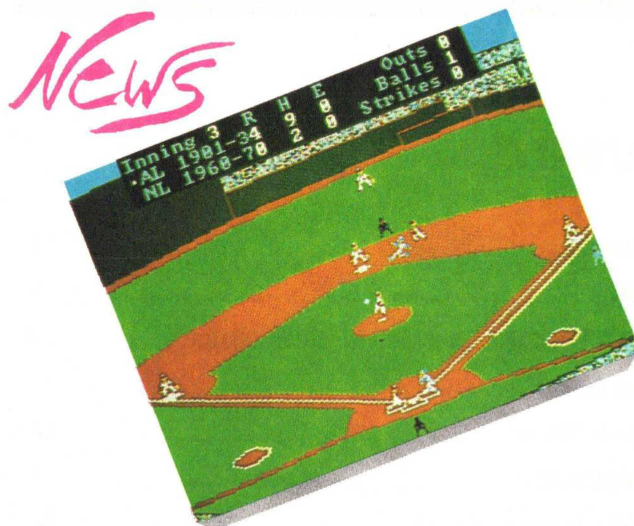


ELECTRONIC ARTS

Das Europabüro von ECA ist umgezogen – raus aus London. Neue Adresse:
Langley Business Centre
1149 Station Road
Langley
NR. Slough
Berkshire SL3 8YN
England
Tel. 00 44 / 753 / 4 94 42.

Neu bei ECA ist eine äußerst realistische Baseballsimulation. Bereits kurz nach seinem Erscheinen wurde EARL WEAVER BASEBALL von amerikanischen Fachzeitschriften zum „Spiel des Jahres" vorgeschlagen.

News





MICRO ILLUSIONS

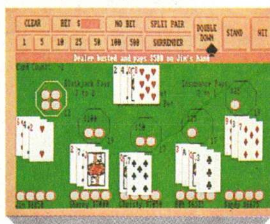
Bei Micro Illusions rollt die Softwarewelle. Für dieses Jahr sind noch mehr als 30 Titel angekündigt. Der Schwerpunkt liegt bei Educational und Professional Software. Aber auch die mit FAERY TALE ADVENTURE so eindrucksvoll begonnene Spielserie wird fortgesetzt.

August:

Educational Software: The Planetarium – ein lehrreicher und unterhaltsamer Blick auf unser Universum. Mehr als 9000 Sterne und Planeten. Discovery Data Disk/Science. Inhalt: Biologie, Chemie, Physik und andere mehr. DDD/Mathematics. Inhalt: Geometrie, Trigonometrie, logische Probleme und anderes mehr. DDD/Trivia 1. Inhalt: Fragen zu den Kategorien Sport, Nobelpreise u.a.m. DDD/Trivia 2. Inhalt: Fragen zu den Kategorien Filme, Raumfahrt, bekannte Persönlichkeiten u.a.m. Math 1. Inhalt: Grundwissen der Mathematik. Spelling 1. Inhalt: Grundwissen der Rechtschreibung.

Spiele:

Fire Power – das erste Spiel einer Reihe, die sich „One to One series" nennt. Der Stil entspricht den Arcade-Games. Man kann alle Spiele dieser Reihe entweder allein gegen den Computer, gegen einen zweiten Spieler (geteilter Bildschirm) oder PER MODEM gegen den Besitzer eines anderen Rechners spielen. Dies ist im Grunde zwar scheißegal, aber einige Leute benötigen eben das Extravagante um glücklich zu werden (Zitat: Harald Schneider). Fire Power ist eine Panzerschlacht mit schnellem Scrolling, guter Graphik und digitalisiertem Sound. Black Jack Academy – ein unterhaltsames Spiel- und Lernprogramm für Kartenhaie und Anfänger. Es ist das erste in einer Serie von Spielen mit dem Thema „Casino". Romantic Encounters at the Dome – ein Spiel nur für Erwachsene. Der Dome ist einer der bekanntesten Nachtclubs Amerikas, und dem Besucher stehen allerlei Abenteuer bevor.



September:

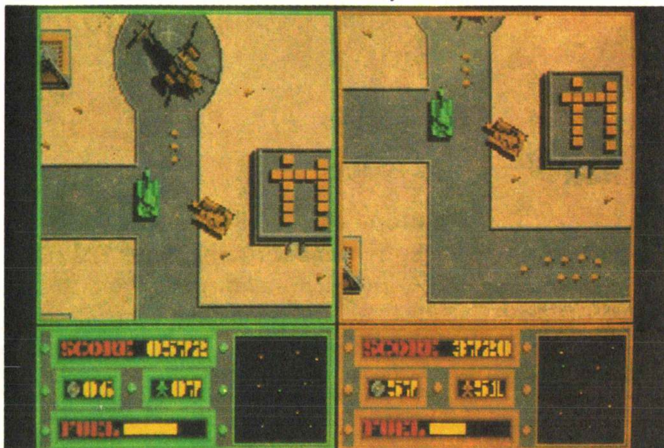
Educational Software: DDD/Geography. Inhalt: USA und Weltkarte, ökologische und wirtschaftliche Beikarten etc. Spelling 2. Inhalt: Rechtschreibung – zehn Stufen mit wachsendem Schwierigkeitsgrad

Professional Software: Dynamic Word – ein Textverarbeitungssystem mit einem Wortschatz von mehr als 100.000 Worten. Die Erstausgabe ist in Englisch, eine deutsche Version soll später erscheinen. Die Vorankündigung verspricht sehr viele Optionen. In einer unserer nächsten Ausgaben werden wir das Programm testen. Music X und Micro Midi – zwei Programme im Bereich Musik, über die im Moment noch nichts näheres zu erfahren ist.

Spiele: Land of Legends – Liebhabern von Dungeon- und Rollenspielen bietet Micro Illusion eine eigene Reihe von QUESTMASTER Games an. Sie beginnt mit Land of Legends und wird noch im Herbst mit dem Dungeon-Constructio-Set fortgesetzt. Turbo – das zweite Spiel der ONE TO ONE-Serie; ein Autorennspiel (Anfang September). Das dritte ist Galactic Invasion – ein Weltraum-Ballerspiel (Ende September).

Für Oktober bis Dezember sind noch folgende Titel angekündigt:

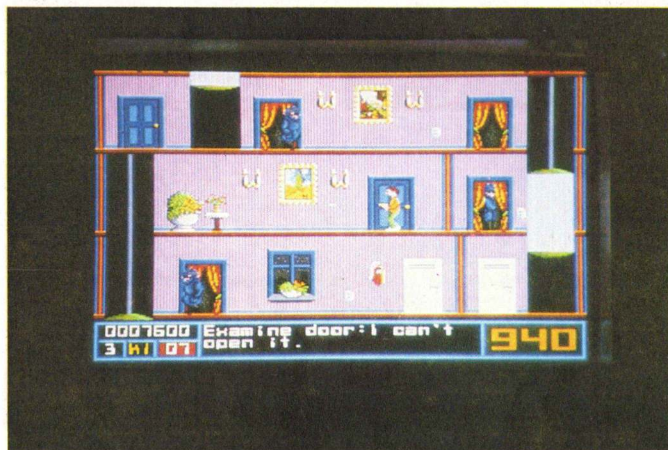
Education: DDD/Language, DDD/History, DDD/Social Studies. Professional: Dynamic CAD 2.4, Photon Video Serie, Dynamic Publisher. Spiele: Ebonstar, TV Shooting Gallery.





KURZ GEMELDET!

Heiße Meldungen, im Bereich Spiele, erreichten uns noch kurz vor Redaktionsschluß. So bringt das Softwarehaus Reline Software ein Spiel namens 'Space Port' heraus, das in etwa dem C64 Klassiker 'Fort Apocalypse' gleichen soll. Kenner wissen vielleicht, daß es darum geht, in eine Art Höhlenlabyrint mit seinem Fluggerät vorzudringen um dort eine Zentrale zu zerstören. Die unterschiedlichsten Gefahren begleiten jedoch den Weg bis dort hin. Die beiden Programmierer, Holger Gehrmann und Uwe Garbosch haben dem Programm etliche Feinheinten verliehen. 'Mission Elivator'aus dem Hause Eurogold, ebenfalls bekannt durch den C64 und dem Atari ST, ist für den Amiga ab sofort erhältlich. Der Spieler muß ein Hochhaus mit etlichen Fahrstühlen durchfahren und Zimmer durchsuchen, in denen unliebsamme Überraschungen warten können. Auf den Gängen bewegen sich dunkle Gestalten, vor denen man sich in acht nehmen sollte. Gute Grafik und digitalisierte Sprachausgabe in deutscher Sprache zeichnen das Spiel aus. Ein luxuriöses Musikprogramm, mit dem Namen 'Dynamic Drums', kommt aus dem Hause New Wave Software. Nicht nur die abwechslungsreiche Aufmachung sollen das Programm auszeichnen, auch Musikfreunde dürfen gespannt sein. Thomas Klinger, Martin und Markus Brenner sind eine dreiköpfige Gruppe von Programmierern die sich 'Chaos Crew' nennt. Die Crew schreibt gerade an drei unterschiedlichen Programmen – 'Space Star', 'Alcazzar' und 'River Raid II'. 'Space Star' ist ein Schießspiel, wobei sich ein Raumschiff mit hoher Geschwindigkeit über eine Landschaft bewegt. Das Demo lies einiges erhoffen. In 'Alcassar' darf ebenfalls geschossen werden, das Scrolling geht hierbei in alle Richtungen und verschiedene Festungen müssen erobert werden. Das dritte Spiel im Bunde nennt sich 'River Raid II' und soll sich auch diesem Spiel annähern.





(AK)

News

Disketten Monitor im Anmarsch!

'DISCovery' nennt sich der neue Disketten Editor aus dem Hause DTM. Die Demoversion, die uns zur Ansicht vorlag, versprach eine ganze Menge. Das Programm ist vollkommen Mausorientiert, sodaß es sehr leicht zu handhaben ist. Sämtliche nur erdenkliche Funktionen wurden oder werden in dem Programm implementiert. 'DISCovery' wird voraussichtlich in den nächsten Wochen erscheinen und wird in Verbindung mit dem deutschem Handbuch 198,– DM kosten. Die jetzt schon erhältliche Demoversion kostet 20,– DM. Ein Ausführlichen Test des Disketten-'Manipulators' wird sobald wie möglich erscheinen. Beziehen kann man die Demoversion, sowie die Discovery unter folgender Adresse:

DTM
Poststraße 25
6200 Wiesbaden-Bierstadt

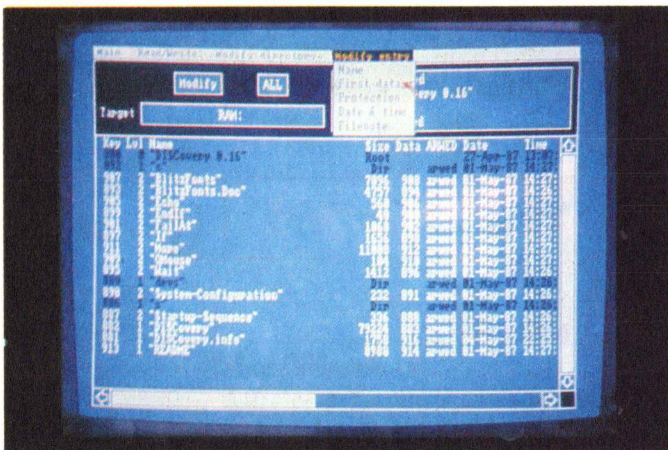
(AK)



Symphony Songs,

unter dieser Bezeichnung vertreibt die Firma DTM eine Reihe von Liedern, die von den Programmen 'Deluxe Music Construction Set', 'Music Studio' und 'Sonix' abgespielt werden können. Die Kollektion beinhaltet momentan 47 Disketten mit den unterschiedlichsten Songs, angefangen von '70's Graetest' über 'Beatles Part 1' bis hin zu 'Classical Part 4'. Jeder Musikgeschmack ist berücksichtigt. Teils sind die Songs nicht ausgereift genug, andere widerum erwecken ein Staunen beim Hörer. Jede Diskette bietet etwa eine Stunde Musik, wobei zwischen 12 und 25 Songs unterschieden werden kann. Der Preis pro Diskette liegt bei üppigen 59,– DM. Nähere Informationen erhalten Sie von:

DTM
Poststraße 21
6200 Wiesbaden-Bierstadt

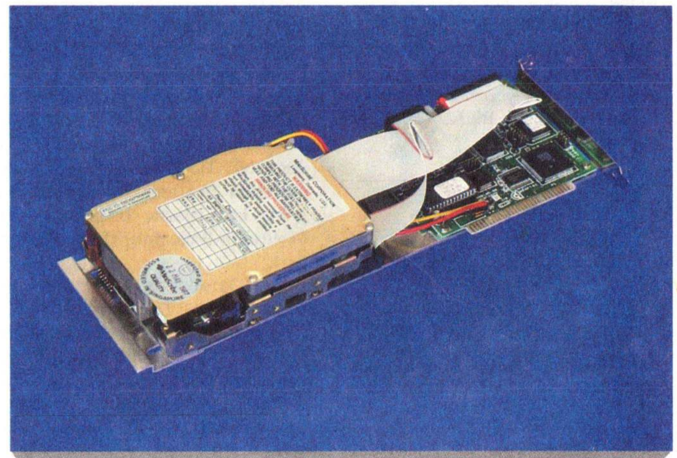


EUROCOMP DiskCard

Die Firma EUROCOMP bietet eine preisgünstige 30 Mega Festplatte für einen PC, XT an. Diese ist demnach geeignet, dem Bridgeboard bzw. der Sidecar zu betreiben. Als Amiga Festplatte läßt sich der Speicherriese allerdings nicht betreiben, da es sich um ein RLL-Aufzeichnungsverfahren handelt. Dieses Verfahren wird leider nicht vom Amiga unterstützt, sodaß der Betrieb nur über die PC-Karte gewährleistet ist. Wer seinen Amiga hauptsächlich unter MSDOS betreibt, dem ist mit dieser DiskCard gedient. Der Einbau in den Amiga 2000 oder die Sidecar geht problemlos und schnell, jedoch verdeckt die DiskCard im Amiga 2000 noch einige zusätzliche Steckplätze, sodaß sie für andere Karten nicht genutzt werden können. In der kurzen Testzeit verrichtete die Karte ihre Arbeit ohne das irgendwelche Schwierigkeiten auftraten. Die technischen Daten der DiskCard: Spannungsversorgung +5 Volt, über PC-Bus +12 Volt, Leistungsaufnahme max. 13 Watt, mittlere Zugriffszeit 60 msec, Übertragungsrate 900 KByte / sec (7.5 MBit / sec), Aufzeichnungsverfahren RLL-Codierung. Der Preis liegt bei 1250,– DM incl. MWSt.

Hersteller und Vertrieb:
EUROCOMP
Dr.-Carl-von-Linde-Str. 21
8000 München 71

(AK)



PEGGY

steht für ein Programm, welches eine Programmieroberfläche für alle Amiga-Systeme darstellt. Mit dessen Hilfe können Programme in beliebigen Sprachen (auch Amiga Basic) erstellt, compiliert und getestet werden. Das Programm stellt hierfür eine Reihe von verschiedenen Bildschirmmasken und einen leistungsfähigen Editor zur Verfügung. Alle nötigen Arbeiten, wie z.B. Kopieren, Löschen und Drucken kann von ‚Peggy' aus gesteuert werden. In der momentan erhältlichen Version 1.1 sind einige Menuepunkte noch ncht implementiert. Version 1.2, die Anfang Januar erscheinen soll, beinhaltet die zum Teil sehr umfangreichen Punkte. Der Preis für das Programm liegt mit 88,— DM recht günstig, das Update schlägt mit 40,— DM zu Buche. Beziehen kann man ‚Peggy' unter folgender Anschrift:

SAS
H.-J. Bernd
5216 Niederkassel 5

(AK)



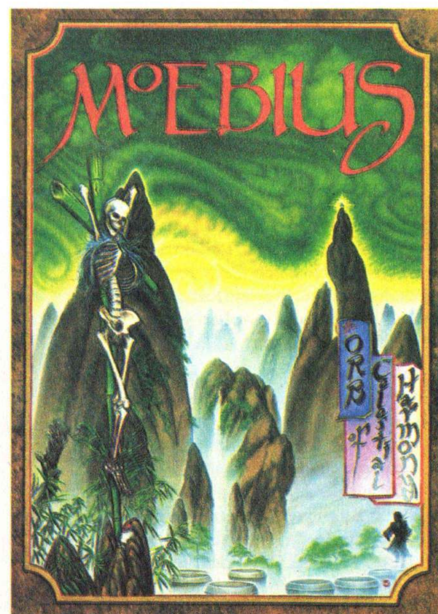
News

Deutsche Anleitung zum Flight Simulator II !!!

Fans des mittlerweilen schon längenderen Flight Simulator II, aus dem Softwarehaus Sublogic, brauchen sich nicht länger mit der englischen Anleitung herumzuschlagen – eine deutsche Übersetzung liegt ab sofort vor. SOFTWARELAND in der Schweiz vertreibt dieses 122 starke Büchlein zum preis von 31,– DM. Bei Interesse dient folgende Anschrift:

SOFTWARELAND
Franklinstraße 27
CH-8050 Zürich (Schweiz)





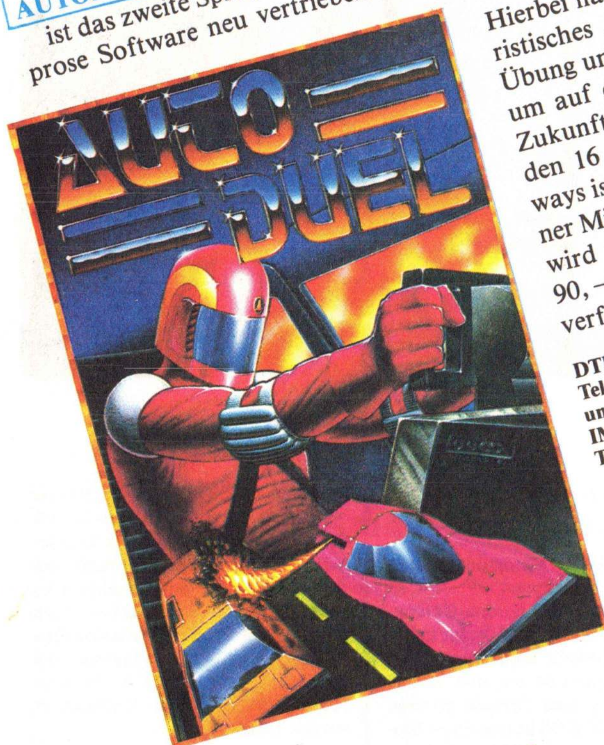
MOEBIUS

nennt sich ein weiteres Rollenspiel für den Amiga, das von der Firma Microprose Software Ltd vertrieben wird. Diese Meldung erreichte uns kurz vor Redaktionsschluß aus dem fernen Amerika. Die Aufgabe des Helden ist es, den gestohlenen Reichsapfel, der sich in den Händen des schrecklichen 'Warlord' befindet, zum rechtmäßigen Besitzer zurück zu bringen. Das Adventure spielt in der orientalischen Welt mit viel Magie, Zauberei und Intrigen. Anfang Oktober wird das Spiel für ca. 90,– DM erhältlich sein. Zu beziehen bei den Händlern:

DTM
Tel. 0 61 21 - 56 00 84
und
IM
Tel. 0 69 - 707 11 02

AUTODUEL

ist das zweite Spiel, das von Microprose Software neu vertrieben wird.



Hierbei handelt es sich um eine futuristisches Auto Strategie Spiel. Viel Übung und Können werden verlangt, um auf den gesetzlosen Straßen der Zukunft zu bestehen. Irgendwo in den 16 Städten und zahllosen Highways ist der Schlüssel zur Lösung seiner Mission versteckt. Das Programm wird etwa Anfang Oktober für ca. 90,– DM bei folgenden Händlern verfügbar sein.

DTM
Tel. 0 61 21 - 56 00 84
und
IM
Tel. 0 69 - 707 11 02



Schluß mit dem Flackern!?

Jeder kennt es, das Flackern des Monitorbildes im Interlace-Modus. Nun gibt es zwei Scheiben, die dem Abhilfe schaffen sollen. Sie sind aus getöntem Kunststoff gefertigt und werden mit Klett-Streifen am Monitor befestigt. Sie sind dadurch leicht wieder abnehmbar. Durch ihre Tönung verstärken die Scheiben den Kontrast, unabhängig vom verwendeten Modus. Allerdings bewirkt die Tönung auch gleichzeitig eine Herabsetzung der Bildhelligkeit, die jedoch weitgehend mit dem Kontrast-Regler des Monitors ausgeglichen werden kann. Von der Wirkung der Scheiben im Interlace-Modus sollte man jedoch keine Wunder erwarten. Zumindest auf dem COMMODORE Model 1081 ist das Flackern so stark, daß auch die Scheiben kein absolut ruhiges Bild erreichen können. Allerdings ist bei beiden Scheiben schon eine sichtbare Beruhigung des Bildes festzustellen. Außer der Tatsache, daß der JIDDER-RID dunkler getönt und mehr als doppelt so teuer ist, konnte zwischen beiden Scheiben kein nennenswerter Qualitätsunterschied festgestellt werden.

(mn)

Nun zu den Daten:

JIDDER-RID ist eine dicke, ziemlich dunkel getönte Scheibe mit abgerundeten Ecken. Sie ist speziell für den COMMODORE 1081 konzipiert. Der Preis liegt bei 79 DM . Der FLICKER MASTER ist der billigere der beiden Kandidaten. Sein Scheibe ist dünner und nicht so dunkel getönt. Sie ist etwas größer und paßt auf deshalb auch noch auf den MAGNAVOX RGB 80 und den SONY KV-1311 Monitor. Der Preis beträgt 35 DM.

Bezugsquelle für beide Scheiben:
PDC
Bad Homburg

Vier Tage wehten Commodore Fahnen vor dem Holiday Inn auf Malta

Anlaß war das diesjährige Commodore Presseseminar zu dem 38 ausgewählte Journalisten und Redakteure der unterschiedlichsten Zeitschriften eingeladen waren. Was es dort zu sehen und zu hören gab, wollen wir Ihnen nicht vorenthalten.

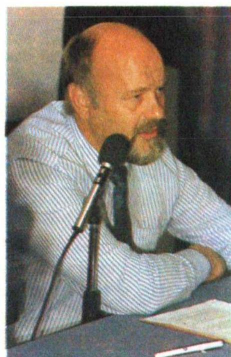


War es letztes Jahr die Kanareninsel Fuerteventura, so lag diesmal die kleine Mittelmeerinsel Malta in der Gunst der Organisatoren. Der Grund für ein Presseseminar außerhalb Deutschlands liegt klar auf der Hand, zum Einen ist es für alle Beteiligten eine angenehme Abwechslung das „Geschäftliche" mit Sonne und Meer zu verbinden, so daß die Teilnehmerzahl entsprechend groß war, zum Anderen, und das ist wohl für den Veranstalter der wichtigere Grund, ist der Zusammenhalt der Gruppe für die gesamte Zeit gewährleistet. Es kann also keiner, der oft so hektische Redakteure, schon nach dem ersten Vormittag aus welchem Grund auch immer, in seine Redaktionsräume zurückkehren.

Doch was gab es vom 22. bis 26. August auf Malta zu hören? Neben den neuesten Verkaufszahlen wurden die Zielgruppen der Amiga Anwender herauskristallisiert. Aber vor allem wurde das ständig wachsende Softwareangebot und die immer bessere Qualität der Programme, darunter besonders die gerade in letzter Zeit steigende Anzahl von Anwenderprogrammen, gelobt. In Sachen Hardware wurde der nun in Produktion befindliche PC-Clone — der PC 1 — Commodores kleinster MS-DOS Rechner präsentiert und vorgeführt.

Geschäftsführer Winfried Hoffmann zieht Bilanz

„Der Amiga hat es geschafft", so die einleitenden Worte von Winfried Hoffmann. Über 60 000 Amigas sind bereits in Deutschland verkauft und noch vor Jahresende soll der 100 000ste Amiga über die Ladentheke wandern. Insgesamt hat Commodore Deutschland im Geschäftsjahr 86/87 (1.7.86 bis 31.6.87) 720 000 Computer verkauft, und damit einen Umsatz von 517 Millionen Mark erwirtschaftet, was einer Steigerung von 100 Millionen oder 22 Prozent gegenüber dem Vorjahr entspricht. Damit sind laut einer Statistik 54 Prozent aller deutschen Computer-Anwender mit einem Commodore Rechner beliefert worden, was zugleich Marktführer in Stückzahlen bedeutet. Um eine ähnlich positive Bilanz auch im laufenden Geschäftsjahr zu erlangen, hat Herr Hoffmann für das Herbstgeschäft eine Werbekampagne in noch nie dagewesener Größe mit Schulungen, Seminaren und den Medien angekündigt, wodurch ein regelrechtes „Amiga Fieber" ausbrechen soll. Schließlich seien erst 14 Prozent des Schul- und Ausbildungswesens, 21 Prozent des Homecomputer Marktes und 32 Prozent des kommerziellen Marktes abgedeckt, so daß noch reichlich Platz zum Expandieren bleibt. Abschließend eine Zahl zum Vergleich: Von Commodores Spitzenreiter, dem C64, sind insgesamt allein in Deutschland fast 1.300.000 Stück verkauft worden!



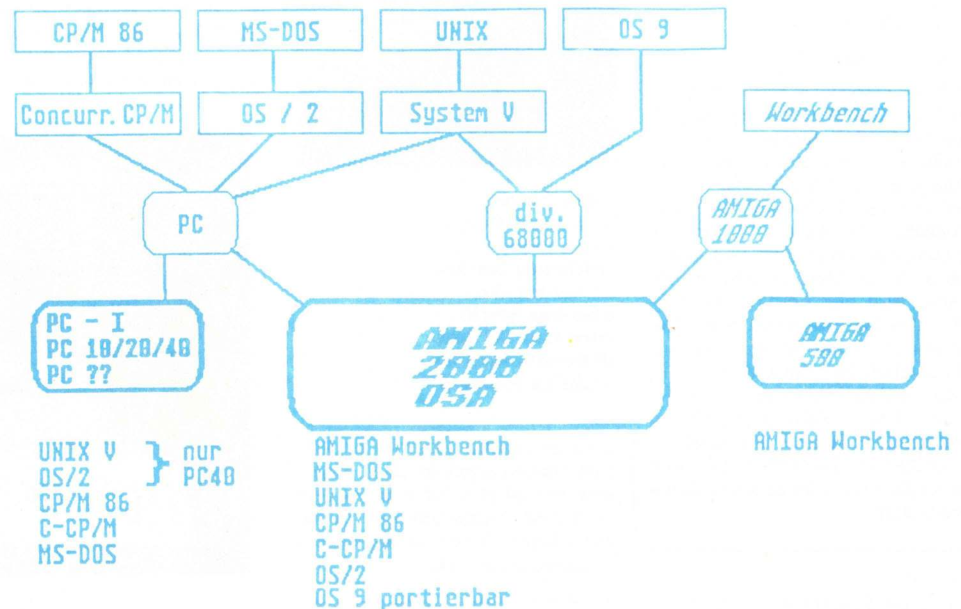
Die Zielgruppe der Amiga Rechner

Der Vertrieb der Commodore Rechner ist in zwei Gruppen unterteilt, in den Home- und den System-Computer Bereich. Zur ersten Gruppe gehören alle Modelle angefangen vom C16 über den C64 bis zum Amiga 500. Zum System Bereich gehören die reinen MS-DOS kompatiblen Modelle PC1, PC10, PC20 und der PC40 und an deren Spitze natürlich der „Alleskönner", der Amiga 2000. Die Bedeutung der Amiga Modelle, die jeweils die Topmodelle der beiden Vertriebsbereiche bilden, läßt auch die Zielgruppe dieser Rechner erahnen. Übrigens wird der Amiga 1000 zur Zeit nicht mehr produziert, ob die Produktion überhaupt nocheinmal aufgenommen wird, ist unklar, eventuell bekommt er das gleiche Innenleben (sprich Platine) wie der Amiga 500. Obwohl Commodore noch den größten Umsatz mit seinen Homecomputern macht, ist ein Trend zu höherwertigen Computern in Aussicht. Durch dieses qualitative Wachstumsdenken wird der Amiga 500 als Nachfolge- bzw. Umsteiger-Modell der C64 und 128er Besitzer angesehen. Mit dem Amiga 2000 steht dem Anwender eine ernstzunehmende, zukunftssichere und ausbaufähige Workstation für alle professionellen Anwendungen zur Verfügung, die durch die offene System-Architektur (OSA) nach Belieben konfiguriert werden kann.

Erweiterungen für den Amiga 2000

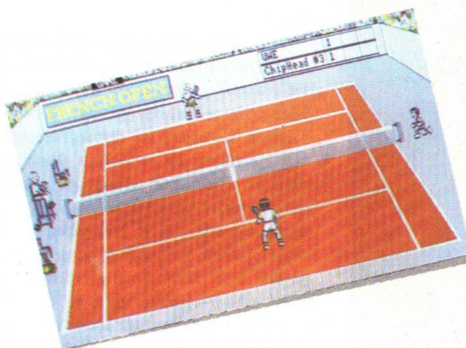
Abbildung 1 zeigt gängige Betriebssysteme für 16/32-Bit Systeme und die entsprechenden Commodore Rechner, auf denen diese eingesetzt werden können. Deutlich hebt sich der Amiga 2000 durch seine offene Systemstruktur von der reinen „Intel-Welt" auf der einen Seite und einer reinen „Motorola-Welt" auf der anderen Seite ab. Der Amiga 2000 kann durch den Einsatz verschiedener Steckkarten die meisten Betriebssysteme verarbeiten, er verbindet sozusagen als einziger Rechner diese „Welten".

Existiert bereits eine PC-Karte mit dem Intel Prozessor 8088 für den Amiga 2000, so wird es zur Systems '87 in München höchstwahrscheinlich auch eine AT-Karte mit dem 80286 Prozessor geben (auf Malta war sie noch nicht zu sehen). Wagen wir noch einen Blick in die Zukunft. Für den Amiga 2000 wird es eine Turbo 2000 CPU Karte geben, die den echten 32-Bit Prozessor von Motorola 68020 samt mathematischen Coprozessor 68881 beherbergen wird. Diese Karte wird in zwei Ausführungen erscheinen, einmal mit 14 MHz und einmal mit 20 MHz Taktfrequenz. Mit der zuletzt genannten Version kann eine Beschleunigung von bis zu 1445 Prozent erziehlt werden! Desweiteren war von der Entwicklung einer Grafikkarte mit 1024x 1024 Bildpunkten und einer Unmenge darstellbarer Farben zu hören. Auch in Sachen Transputerkarte könnte etwas kommen. Doch nun genug der Spekulationen, begeben wir uns wieder auf den Boden der Realität zurück.



Programmvielfalt nicht nur bei Spielen

Daß es für den Amiga von Anbeginn gute Spiele gab, ist kaum verwunderlich, da er durch seine ausgezeichneten Grafik- und Sound-Eigenschaften dazu prädestiniert ist. Neue Spiele sind „Ferrari Formula One" ein Autorennen, „Grand Slam" ein Tennisspiel mit verschiedenen Courts und ein Baseballspiel, daß seinen Erfolg wohl mehr in den USA suchen wird, da diese Sportart in Deutschland kaum verbreitet ist. Die meisten Spiele der neuen Generation für den Amiga zeichnen sich durch eine hervorragende Grafik und einen fast naturgetreuen Sound (sogar in Stereo) aus. Doch der Amiga ist mehr als nur eine „Super-Spiele-Kiste", das jedenfalls versuchte Commodore durch ein reichhaltiges Angebot an professioneller Anwendersoftware zu beweisen. Natürlich liegen die Stärken dieser Anwendersoftware vor allem im Bereich der Grafik- und Soundprogamme, aber auch Text-, Datei- und Kalkulationsprogramme wurden gezeigt.

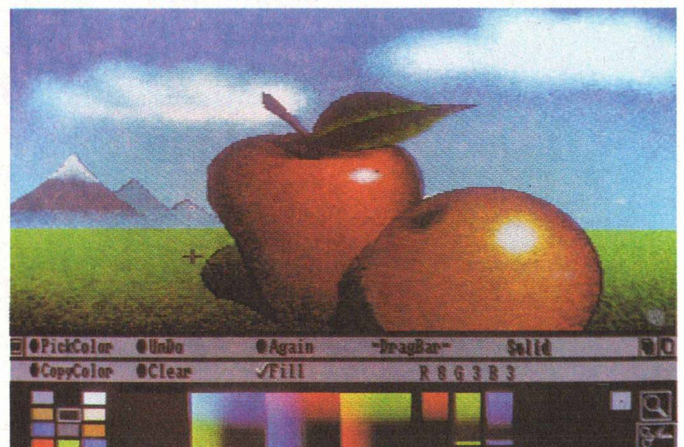


Desktop Publishing

Unter diesem Begriff versteht man das Entwerfen und Gestalten von Texten mit Bildern und Grafiken am Computerbildschirm, so daß ganze Vorlagen oder auch Zeitschriften mit der „Maus" erstellt werden können. Drei Programme existieren bereits: Publisher 1000, City Desk und Page Setter. Mit dem Page Setter und dem Programm Page Setter LaserScript lassen sich die erstellten Seiten auch auf einem Postscript fähigen Laserdrucker, wie z. B. dem LaserWriter Plus von Apple ausdrucken. Insgesamt sind die Leistungen des Page Setters aber nur befriedigend (siehe auch Testbericht in Kickstart Nr. 1 Juni '87). Besseres verspricht das in Kürze erscheinende Programm The Professional Page. Mit diesem Programm wird dann u. a. auch Kerning (Abstand zwischen den Buchstaben wählbar) möglich sein, allerdings wird dieses Programm einen stolzen Preis von etwa DM 800,– haben.

In Sachen Grafik...

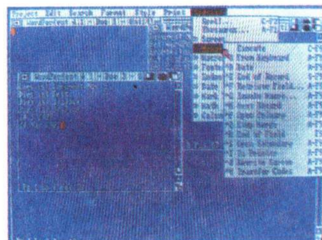
gibt es mittlerweile wirklich ausgezeichnete Programme, die teilweise in amerikanischen Filmstudios zum Einsatz kommen. Mit den Programmen „Provideo" oder „Desktop Video" lassen sich Werbespots, animierte Bilder oder Vorspänne für eigene Videofilme herstellen. Mit Desktop Video lassen sich auch ganze Trickfilme produzieren. In die gleiche Kategorie gehören auch „Video Titler" und „Videoscape 3D" von Aegis. Letzteres wird an anderer Stelle in diesem Heft ausführlich vorgestellt und getestet. Zu einer anderen Gruppe gehört „Sculp 3D", das in den News dieses Heftes kurz vorgestellt wird. Überzeugend wurde ein neues Malprogramm namens „Digi Paint" von NewTek vorgeführt. Mit diesem Programm lassen sich 4096 Farben gleichzeitig auf dem Monitor darstellen. Digi Paint arbeitet im 320x200 und im hochauflösenden 320x400 Punkte „Hold and Modify (HAM)" Modus. Besonders Eindrucksvoll ist die Möglichkeit vorhandene schwarzweiß Bilder (IFF-Standard) in Farbbilder zu verwandeln. Vorhandene Grautöne lassen sich nicht nur durch eine Farbe ersetzen, vielmehr können sie mit einer Farbe logisch „verknüpft" werden, so daß weiche, sehr realistische Farbübergänge entstehen. Digi Paint werden wir in der nächsten Kickstart ausführlich testen.



Digi Paint – bis zu 4096 Farben gleichzeitig

Text, Datei und Kalkulation

Word Perfect, ein verbreitetes und beliebtes Textverarbeitungsprogramm auf MS-DOS Rechnern, wurde in der englischen Version nun auch für Amiga gezeigt. Die deutsche Version soll Anfang Oktober auf den Markt kommen. Die Amiga Version, die größtenteils sowohl über die Tastatur als auch die Maus bedient werden kann, wird alle Features der MS-DOS Version beinhalten. Superdesk ist das Programm für den täglichen Bürobedarf. Es beinhaltet ein Text-, Datei- und Kalkulationsprogamm und wird auf neun Disketten geliefert, so daß eine Festplatte unbedingt zu empfehlen ist. Im Lieferumfang sind einige Bücher für die einzelnen Bereiche enthalten.



Word Perfect – mehrere Dokumente können gleichzeitig bearbeitet werden. Verschiedene Schriftarten sind auf dem Bildschirm auch als solche zu erkennen.

Fremde Hardware

Am Rande der Veranstaltung konnte ein nachleuchtender Farbmonitor von Mitsubishi bewundert werden. Das Modell (C-3479 ELPAC) zeigte durch die nachleuchtende Bildröhre selbst im Interlace Modus am Amiga ein fast stehendes Bild. Für lange Arbeiten am Bildschirm kann ein solcher Monitor nur empfohlen werden. Ob Commodore diesen oder einen ähnlichen Monitor in die Produktpalette aufnimmt, war nicht zu erfahren.

Überzeugend waren auch die Farbausdrucke des PlotMasters der Firma Calcomp in Düsseldorf. Dieser nach dem Thermo-Transfer-Verfahren arbeitende Drucker kann bis zu 4000 Farbschattierungen in einer Auflösung von 203,2 x 200 dots per inch (= 8x7,87 Punkte pro mm) zu Papier bringen. Große Flächen werden gleichmäßig bedruckt.

Commodores kleinster MS-DOS Rechner – der PC1

Der PC1 rundet Commodores Produktpalette der PC-Clones nach unten ab. Er ist somit als preiswerter Kompatibler für den MS-DOS-Einsteiger gedacht. Durch die Verwendung hochintegrierter Custom-Chips konnte ein sehr kompakter Rechner mit den Eigenschaften eines Original IBM-PCs realisiert werden. Die technischen Daten des PC1 finden Sie in der Tabelle. Passend zum PC1 gibt es eine sogenannte Erweiterungsbox, die über ein eigenes Netzteil und drei PC kompatible Slots in voller Länge verfügt. In dieser Expansion-Box ist ferner Platz für eine 5 1/4 Zoll Einheit, so daß eine zweite Floppy, die unter MS-DOS eigentlich nur zu empfehlen ist, oder auch eine Festplatte in diese Box eingebaut werden kann.



Technische Daten: PC1

CPU
Prozessor: 8088
Taktfrequenz: 4,77 MHz
Sockel für Coprozessor 8087 vorhanden

Speicher
Standard RAM: 512 KByte
intern erweiterbar auf: 640 KByte

Massenspeicher
eingebautes Diskettenlaufwerk:
5 1/4" DS/SD
Kapazität: 360 KByte

Bildschirmdarstellung
Monochrome:
80x25 Zeichen
720x348 Punkte (Hercules)

Farbe:
40x25 oder 80x25 Zeichen, 16 Farben
320x200 Punkte, 4 Farben (CGA)
640x200 Punkte, 2 Farben (CGA)

Anschlüsse
Parallel

RS 232, seriell

Maus-Port (zu Amiga kompatibel)

23 pol. Sub-D Buchse für ein Zusatzlaufwerk (Amiga kompatibel)

RGB und Comp.-Video Ausgang

60 pol. Platinenstecker für Expansion-Box

5 pol. PC-Tastatur Buchse

Netzteil
eingebautes 50 Watt Schaltnetzteil

Tastatur
84 Tasten im AT-Look

Preis
DM 1298,–
(ohne Monitor und Maus)

Fazit

Weltbewegende Neuigkeiten waren auf Malta nicht zu sehen, selbst der PC1 wurde ja schon auf der CeBit im März dieses Jahres gezeigt. Und in Sachen Software ist der aufmerksame Kickstart Leser schon über die meisten Produkte informiert. Aber Commodore ging es wohl auch gar nicht darum, viele Neuigkeiten zu zeigen, denn schließlich ist der Amiga 500 bzw. 2000 noch nicht mal ein Jahr alt. Vielmehr ging es darum, daß der Amiga immer mehr „Freunde“ gewinnt, sowohl bei den Anwendern als auch den Soft- und Hardwarehäusern, die Produkte für ihn produzieren. Doch, ob der Amiga es schon geschafft hat, besonders im professionellen Markt, bleibt immer noch abzuwarten.

(UB)





RAY-TRACER selbst gemacht – SCULPT 3D

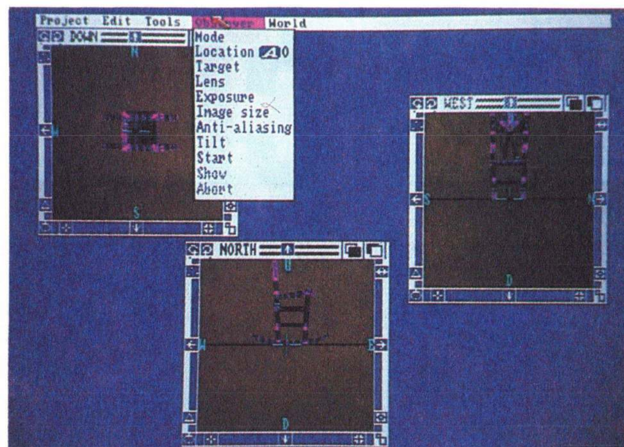
Jeder kennt die DEMO des Männchens, das mit drei verspiegelten Kugeln jongliert. Ab sofort ist ein Programm erhältlich, mit welchem man auf dem Amiga Ray Tracer-Bilder erstellen kann. Die Gittermodelle der Objekte werden mit einem leicht zu bedienenden Editor erstellt. Das Programm übernimmt das Füllen der Flächen in Schattierungen und Features zur Darstellung von Körperlichkeit. Die dafür benötigten Berechnungen sind derart komplex, daß unter Umständen mehrere Stunden dabei vergehen.

Es entstehen faszinierende Körper, die noch in Bezug auf Farbgebung, Blickwinkel, Beleuchtungsrichtung usw. bearbeitet werden können.

Hersteller: BYTE by BYTE

Preis: DM 195,–

Vertrieb:
IM 0 69 / 707 11 02
DTM 0 61 21 / 56 00 84
PDC 0 62 72 / 2 47 48



SILVER RAY TRACE ANIMATOR

Ebenfalls angekündigt ist ein RAY TRACE ANIMATOR, mit dem die RAY-TRACER-Bilder in Bewegung versetzt werden können. Hersteller ist die amerikanische Firma IMPULS, die gerade DIGI VIEW 2.0 herausgebracht haben. Der Objekt-Editor läßt sich mit der Tastatur und der Maus bedienen. Daraufhin wird die Beleuchtungs- und Aufnahmerichtung festgelegt. Die Berechnung erfolgt innerhalb von 4 Minuten. SILVER arbeitet in allen Bildschirmauflösungen des AMIGA. Die Abspielgeschwindigkeit beträgt zwischen 6 und 60 Frames pro Sekunde. Genlock und Overscan werden unterstützt. Preis: ca. DM 349,–

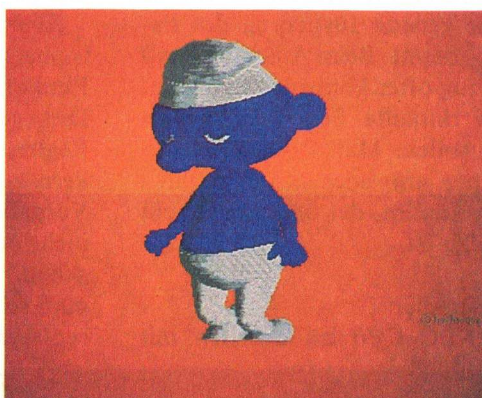
FORMS IN FLIGHT

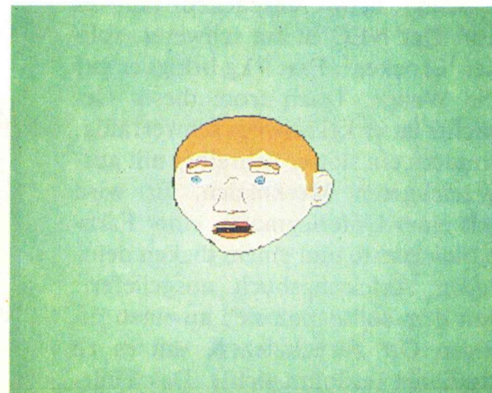
Ist ebenfalls ein 3D Programm zum Zeichnen und Animieren. Es läuft ausschließlich im Interlace-Modus und benötigt mindestens 1 MB Speicher. Es ist sowohl fürs Hobby als auch für ernsthafte Anwendungen gedacht. Eine interessante Option ist der Rot/Blau-Modus, der es erlaubt räumlich erscheinende Objekte zu erstellen. Wie David Youlton von MICRO MAGIC zusicherte, soll ab Anfang September auch eine PAl-Version erhältlich sein. Diese Version wird exclusiv von INTELLIGENT MEMORY vertrieben, die zusätzlich eine deutsche Anleitung und eine rot/blau-Brille mitliefern. Der Preis wird bei 165 DM liegen.

Bezugsquelle:
IM 0 69 / 707 11 02
DTM 0 61 21 / 56 00 84

DISNEY 3D ANIMATOR

Der erste Trickfilmgenerator ist da! Der Name verspricht viel. Ein ca. 120-seitiges deutsches Handbuch wird zur Zeit erstellt und mitgeliefert. Für 20 DM kann man zur Zeit eine Demo-Diskette erwerben, die einen animierten Schlumpf zeigt. Der Preis für den DISNEY 3D ANIMATOR beträgt 595 DM. Für den kleineren Geldbeutel gibt es außerdem noch eine Junior-Version, die 189 DM kosten wird. Als Zusätze sind mehrere Disketten mit SHAPES zu Preisen zwischen 79 DM und 89 DM erhältlich.







Ein Drucker für alle Fälle: NEC CP6



Die Firma NEC hat mit ihrer Pinwriter-Serie einen wahren Volltreffer gelandet. Der NEC P6 hat in der Sparte der 24-Nadel-Matrixdrukker die Vorherrschaft von EPSON durchbrochen und ist zu einem Quasi-Standard geworden. Die Farbversion dieses Druckers steht diesem in nichts nach. Im Gegenteil – sie hat etwas mehr...

Bedienung

Schon beim Auspacken fällt es auf: Der NEC ist ein schwerer, solider 'Brocken'. Fast 9 kg bringt er auf die Waage. Doch trotz dieses Gewichts ist er keineswegs schwerfällig, sondern ein flotter Schreiber mit ausgezeichneten Merkmalen. Er wird mit einer Unterlegmatte, einer Farbbandkassette und einem dicken deutschen Anleitungsbuch ausgeliefert. Mit dem sollte man sich an einen ruhigen Ort zurückziehen, um es zu studieren (es lohnt sich!). Das Einlegen der Farbbandkassette ist nicht schwer und man bekommt keine dreckigen Finger, denn zwei Handschuhe werden mitgeliefert. Der Einzug des Einzelpapiers erfolgt halbautomatisch. Endlospapier kann ebenfalls verwendet werden, allerdings ist kein Traktor vorhanden, so daß es auf eine genaue Justierung des Papiers ankommt. Beim Ausdruck muß der Einzug öfter kontrolliert werden, wenn zerknäulte Seiten vermieden werden sollen. Man kann allerdings auch einen uni- oder bidirektionalen Traktor kaufen, der allerdings nicht ganz billig ist.

Druckertreiber

Der NEC CP6 ist identisch mit dem COMMODORE MPS 2000 C, dadurch kann er dessen Druckertreiber von der Workbench benutzen. Dieser ist mit einer Länge von rund 10 KB fast doppelt so groß wie die übrigen. Nicht zuletzt daran zeigt sich, daß sich die Programmierer hier die meiste Mühe gegeben haben, um auch wirklich das Letzte aus dem Drucker herauszuholen. Daß ihnen dies tatsächlich geglückt ist, zeigt sich an der hervorragenden Qualität der Ausdrucke. Die 24 Nadeln und die hohe Punktdichte werden vollständig ausgenutzt, um die hochauflösende und brillante Grafik des Amiga so gut wie möglich wiederzugeben (siehe Bild 1). Das Ergebnis ist dabei immer wieder erstaunlich. Kein anderer Drucker dieser Preisklasse schafft diese Farbbrillanz und solchen Kontrastreichtum. Dies zeigt sich vor allem bei Schwarzweißbildern, die von Schattierungen und Körperlichkeit leben (siehe Bild 3).

Eine Besonderheit dieses Druckertreibers ist es außerdem, daß die Punktdichte eingestellt werden kann. Dies geschieht über den Tresholds-Regler, der sonst nur bei s/w-Drukkern zum Einsatz kommt. Je nach Vorgabe werden die Farben dann mehr oder weniger kräftig wiedergegeben. Im unteren Bereich erhält man sehr schwache Farben, im oberen werden sie dagegen zu kräftig

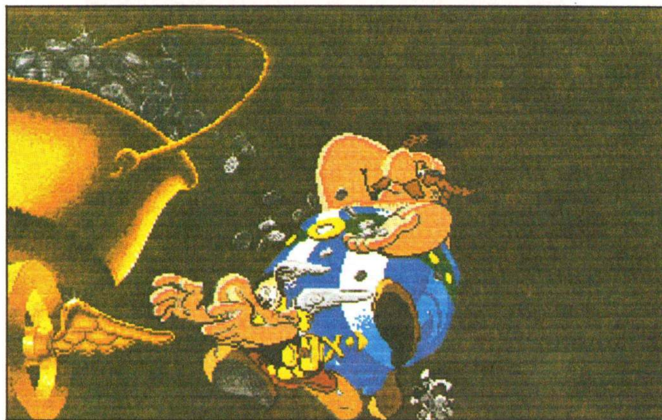


Bild 1: Brillanter Farbausdruck

und dunkel. Als günstig hat sich ein Wert von 9 – 10 erwiesen. An dieser Stelle soll jedoch nicht verschwiegen werden, daß diese Punktdichte auch ihren Preis hat: Zwischen 30 und 45 Minuten braucht der NEC dann schon, um ein Bild zu vollenden.

ohne ihn auszukommen. Wer das Geld für den CP6 aufbringen kann, dem kann nichts besseres empfohlen werden. Es wird danach sicherlich keine Reue aufkommen, denn der Gegenwert ist sehr solide und überzeugend.

(mn)

+ schnelle Geschwindigkeit bei Draft und NLQ
+ sehr guter Grafikausdruck
+ viele Schrifttypen, Proportionalschrift
+ Zeichensatz ladbar
+ leistungsbezogen günstiger Preis

– langsamer Grafikausdruck
– Traktor nur optional und umständlich in der Handhabung

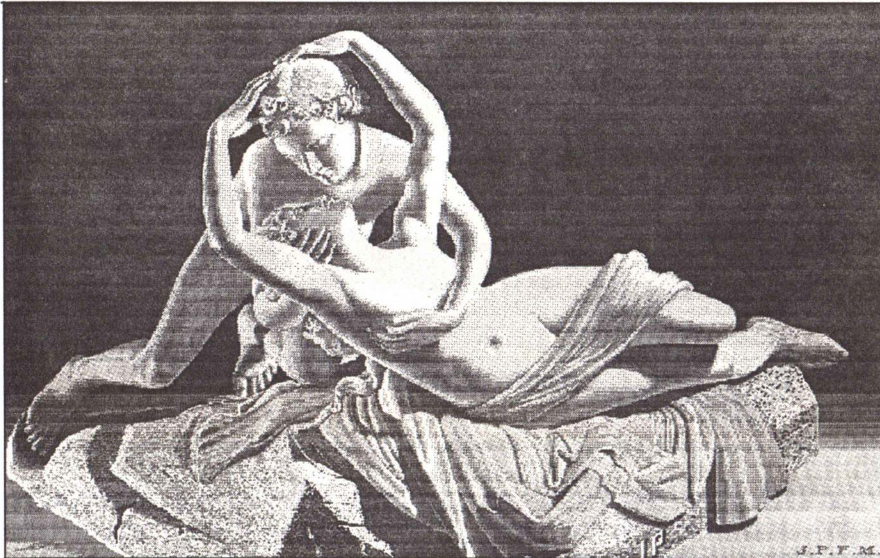


Bild 3: Kontrastreiches S/W-Bild

Wertung

Viel bleibt nicht mehr zu sagen. Der NEC CP6 gehört sicher zu den schillerndsten und interssantesten Druckern für den Amiga. Dies liegt vor allem an seiner sehr sauberen Schönschrift und der für Papierdruck brillanten Farbwiedergabe. Etwas geschmälert wird diese Begeisterung nur durch den fehlenden Traktor bzw. dessen umständliche Handhabung. Es ist aber durchaus möglich, sich damit zu arrangieren oder auch

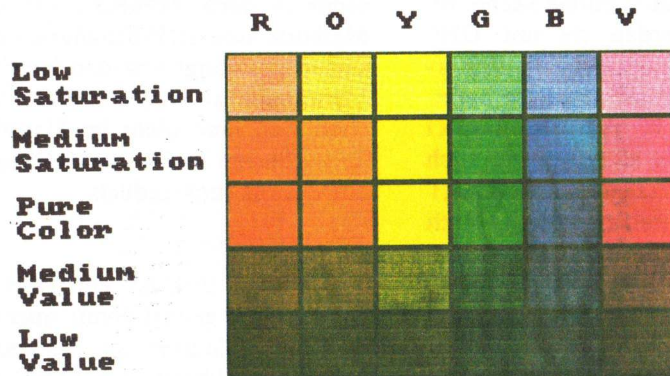


Bild 2: Farbpalette

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Normalschrift
Fettschrift
Doppeltdruck
Unterstrichen
Breitschrift
Schmalschrift
Superscript
Subscript

Bild 4: Einige Schriftproben des CP6

NEC CP6

Daten:
Typ: 24 Nadel Matrixdrucker (bis zu 360x360 Punkte/Zoll)
Hersteller: NEC
Kompatibel: MPS 2000 C
Druckertreiber: ja
Preis: ab 1500,– DM

Geschwindigkeit:
Draft: 180 Zeichen/sec.
NLQ: 60 Zeichen/sec.
Grafik: 1 – 28 Min. (Stufe 1 – 10) (je nach Druckdichte)

Besonderheiten:
12 internationale Zeichensätze
8 KB Puffer
leise (56/53 dB)
deutsches Handbuch

Optional:
(halb-)autom. Einzelblatteinzug
uni-/bidirektionaler Traktor

Der kleine Bunte: OKI 292

Der OKI 292 präsentiert sich als zierlicher Drucker mit einem durchaus ansprechenden Äußeren. Die Walze mit dem integrierten Traktor erlaubt es, ohne Umbau Einzelblätter und Endlospapier zu verwenden. Der halbautomatische Papiereinzug und die einfache Druckereinstellung im Dialogbetrieb runden das Bild ab.



Bedienung

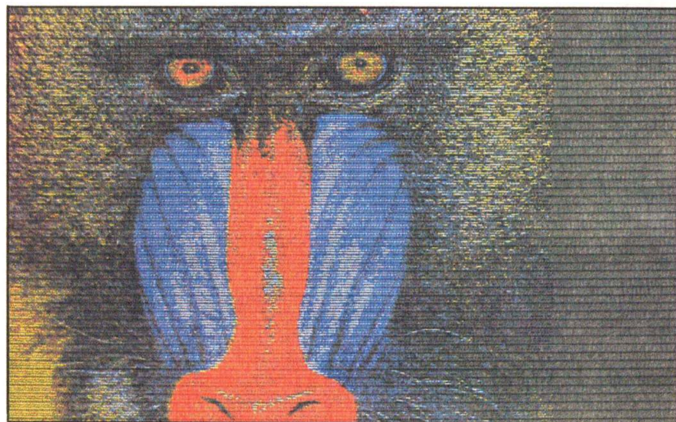
Der OKI ist schnell betriebsbereit: Verbindung zum AMIGA herstellen, Stecker in die Dose, Papier einlegen, Hebel umlegen – fertig! Das Blatt wird (halb)automatisch eingezogen (allerdings zu weit!). Aber auch Endlospapier ist schnell eingespannt, denn die Walze ist mit einem Traktor versehen, der 'genialerweise' ohne Haltebügel auskommt und sogar bidirektional arbeitet (meiner Meinung nach eine sehr gute Lösung!).

Auch wenn es sicherlich nicht gleich nötig ist, so sind doch die Druckereinstellungen eine wichtige Sache. Im allgemeinen werden sie mit DIP-Schaltern vorgenommen, die sich irgendwo im oder am Gehäuse befinden. Diese sucht man beim OKI vergeblich, denn er wird, wie auch der MPS 1500 C (siehe KICKSTART 9) im Dialogbetrieb eingestellt. Nach der Aktivierung wird eine Frage ausgedruckt, z. B. PRINT MODE NLQ? Ist man mit dieser Einstellung einverstanden, dann wird FORM FEED gedrückt, andernfalls die Taste SELECT. Nun ist der Schnellschriftmodus eingestellt. Auf die gleiche Weise werden noch ca. 30 weitere Fragen durchgespielt. Dies dauert schon etwas und benötigt nicht gerade wenig Papier. Gottseidank bleibt diese Einstellung jedoch auch nach dem Ausschalten noch erhalten. Ob diese Methode den DIP-Schaltern vorzuziehen ist, hängt von den jeweiligen Umständen ab. Sie ist sicher einfacher, weil man nicht im Handbuch nachschlagen muß, aber dafür ist sie auch etwas umständlich.

Druckertreiber

Der OKI 292 ist ein 18-Nadel-Drucker und gehört damit einer relativ seltenen Gattung an. Dennoch ist auf der Workbench des AMIGA ein entsprechender Druckertreiber vorhanden. Dies ist an sich eine gute Sache, da man sich nicht um die Anpassung bemühen muß, allerdings ist die Arbeitsweise des Treibers nicht gerade optimal. Dies zeigt sich zum einen in der geminderten Qualität der Ausdrucke, zum anderen in der langen Dauer eines Ausdrucks. Stolze 12,5 Minuten benötigt das Gerät für eine Arbeit, die der im letzten Heft vorgestellte COMMODORE MPS 1500 C in nur 7,5 Minuten erledigt (ohne erkennbaren Qualitätsverlust!). Wenn man den Vorgang beobachtet, fällt auf, daß keinerlei Optimierungen vorgenommen wurden, denn der Druckkopf fährt selbst dann die Zeile ab, wenn die gerade eingestellte Farbe gar nicht benötigt wird. Wenn gar eine leere Zeile ansteht, dann wird vier mal im 'Leerlauf' über das Blatt gefahren, ohne daß nur ein Punkt gesetzt wird – wirklich unverständlich!

Den Ausdrucken fehlt es etwas an Brillanz (soweit man davon bei einem Matrixdrucker sprechen kann!). Dies liegt daran, daß der Druckertreiber die Grafikfähigkeiten des OKI nicht voll ausnutzt. Zu wenige Punkte werden gesetzt, wodurch Farbflächen etwas blaß wirken und Schattierung und Körperlichkeit nicht richtig zur Geltung kommen.



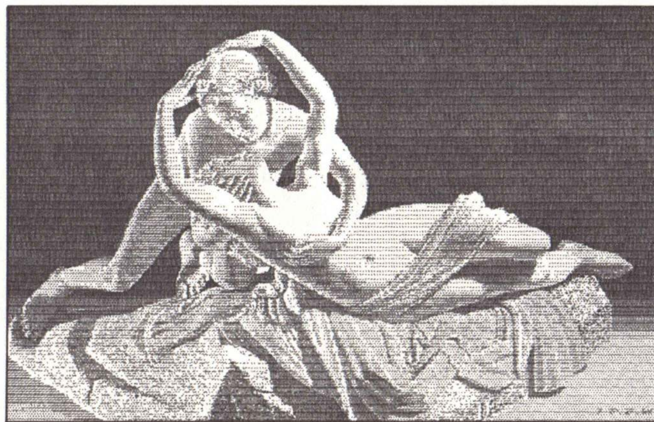
Farbgrafik auf dem OKI

Wertung

Der OKI 292 ist ein kompaktes

Leichtgewicht (nur 4 kg), dessen größter Vorzug die leichte Bedienbarkeit ist. Einzelblatt und Endlospapier sind schnell und einfach eingelegt und lassen sich auch problemlos nebeneinander benutzen. Einziger Kritikpunkt bei der Bedienung sind die Einstellungen im Dialogbetrieb. Dies ist zwar sehr einfach, aber zum Teil etwas umständlich. Die Druckgeschwindigkeit ist sowohl im Schnellschreib- als auch im Schönschriftmodus ziemlich schnell. Das Resultat ist durchaus akzeptabel. Die Qualität der Grafikausdrucke ist annehmbar, sie könnte jedoch besser sein. Schuld daran ist jedoch nicht der Drucker, sondern vielmehr der Treiber der Workbench, der die Fähigkeiten des OKI nicht ausnutzt. Es ist aber anzunehmen, daß es in absehbarer Zeit bessere Druckertreiber gibt. Dann kann der OKI wirklich zeigen, was in ihm steckt. Durch seinen günstigen Preis ist er eine gute Partie, um Texte und Grafiken des Amiga zu Papier bringen zu können.

(mn)



Bei kontrastreichen S/W-Bildern zeigen sich die Schwächen des Treiberprogramms.

R O Y G B V

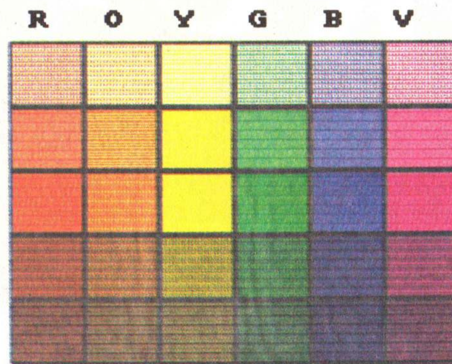
Low Saturation

Medium Saturation

Pure Color

Medium Value

Low Value



Farbpalette

+ hohe Druckgeschwindigkeit
+ guter Taktor
+ einfache Bedienung
+ viele Schrifttypen, Proportionalschrift
+ günstiger Preis

– Einstellungen nur im Dialogbetrieb
– langsamer Grafikausdruck

OKI Microline 292

Daten:
Typ: 18-Nadel-Matrixdrucker
(bis zu 288x288 Punkte/Zoll)
Hersteller: OKI
Kompatibel:
Druckertreiber: ja
Preis: ab 1100,– DM

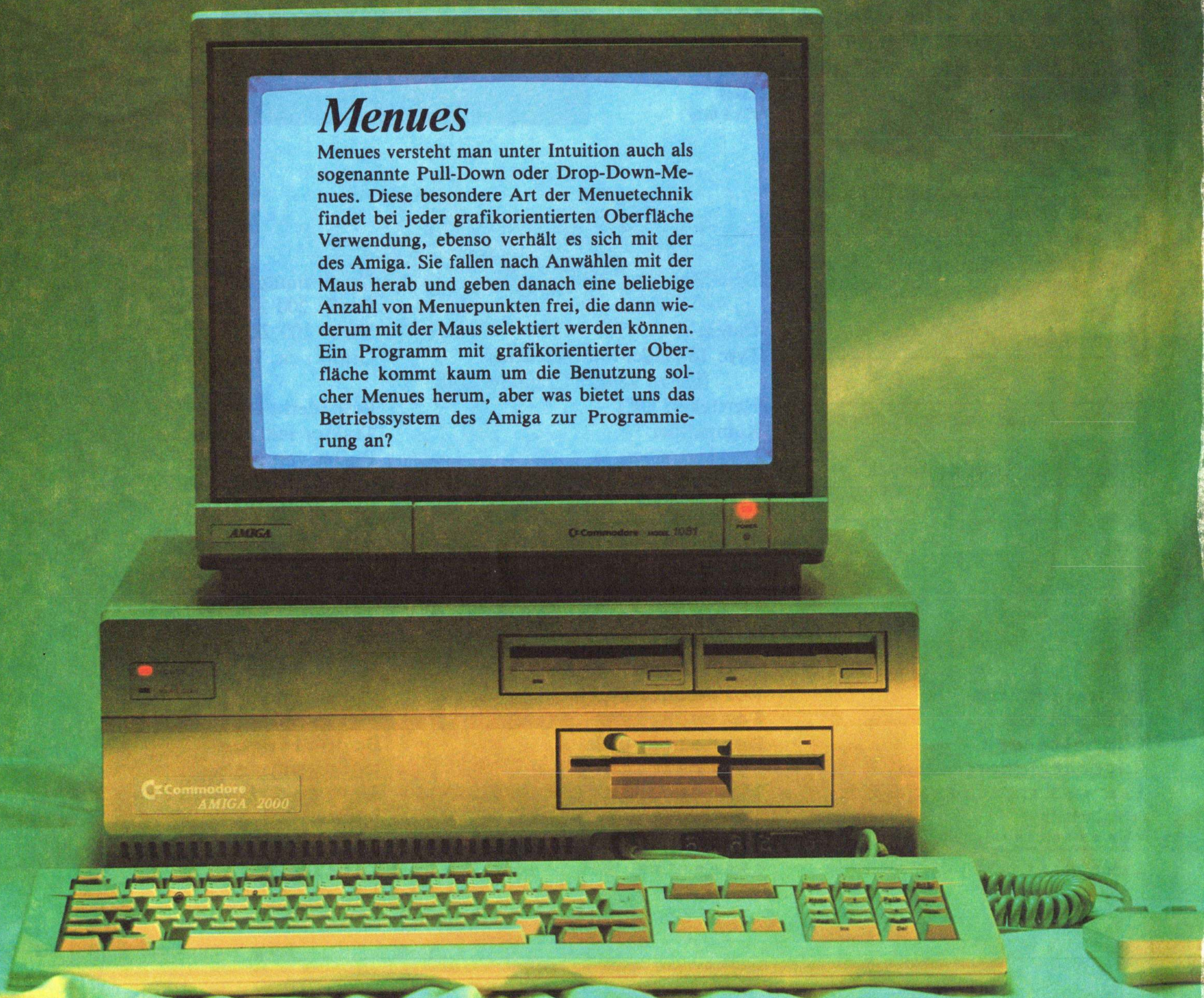
Geschwindigkeit:
Draft: 200 Zeichen/sec.
NLQ: 100 Zeichen/sec.
Grafik: ca. 12:31 min.

Besonderheiten:
Traktor mit integrierter Walze
Puffer von 15 KB
deutsches Handbuch

Normalschrift	Normalschrift	Normalschrift
Fettschrift	Fettschrift	Fettschrift
Doppeltdruck	Doppeltdruck	Doppeltdruck
Unterstrichen	Unterstrichen	Unterstrichen
Breitschrift	Breitschrift	Breitschrift
Schmalschrift	Schmalschrift	Schmalschrift
Superscript	Superscript	Superscript
Subscript	Subscript	Subscript

Einige Schriftproben des OKI 292

INTUITION-KURS 4



Rückblick

Wie immer, möchte ich meinen Kurs mit einem kleinem Rückblick beginnen, so daß auch neue Leser wenigsten einen kleinen Überblick erhalten. Intuition ist eine Bibliothek des Amiga-Betriebssytems. Sie beschäftigt sich hauptsächlich mit der grafikorientierten Benutzeroberfläche, demnach mit Fenstern, Gadgets, Menues, Requestern usw. Ziel des Kurses ist es, Ihnen die einzelnen Komponenten von Intuition etwas näher zu bringen. In den letzten Abschnitten habe ich Bildschirme, Fenster und Gadgets aufgegriffen sowie die Funktionsweise und deren Programmierung erläutert. Dieser Teil befaßt sich nun ausschließlich mit Menues.

Menue-Know-How

Wer kennt sie nicht, diese besondere Art der Menuetechnik: Dem Computeranwender werden nach Anwählen eines Menues eine Vielzahl von weiteren Menueoptionen durch Herunterfallen einer Leiste freigelegt. Diese kann die unterschiedlichsten Formen und Farben aufweisen; man bemerke, daß nicht nur Text, sondern auch Bilder als Menueoptionen verwendet werden können. Bild 1 zeigt die typische Verwendung von Pull-Down-Menues anhand einer Textverarbeitung. Weiterhin können die einzelnen Menuepunkte auch über die Tastatur angewählt werden – was sicher ein geistreicher Gedanke der Entwickler war, denn nicht alle Computeranwender sind von diesen 'neuen' Oberflächen angetan, und manche arbeiten lieber mit der konservativen Methode, der Tastatur. Zweifellos ist nach Einarbeitung in ein Programm die Benutzung der Tastatursequenzen schneller als das Bedienen des Programms über die Maus. Durch Setzen des Flags COMSEQ in der MenuItem-Struktur wird eine Tastatursequenz dem Menuepunkt zugewiesen. Eine solche Belegung setzt sich immer aus der Amiga-Taste sowie einer beliebigen weiteren Taste zusammen, wobei die letztere Taste in der Variablen Kommando festgelegt werden muß. Die Amiga-Taste braucht nicht definiert zu werden, sie wird immer zu diesem Zweck gebraucht.

Doch nun weiter im 'Menuedschungel' des Amiga. In Menuebalken können zwei Arten von Aktionen programmiert werden. Die erste Möglichkeit ist, vom Menue aus einen Befehl mit Hilfe der Maus anzuwählen, beispielsweise das Sichern einer Textdatei in einem Textverarbeitungsprogramm. Die zweite Anwendung ist das Setzen von Programmparametern, z. B. die Baudrate eines Terminalprogramms. Im ersten Fall wird die Anweisung ausgeführt und in das Programm zurückgekehrt, während die zweite Möglichkeit nur den Parameter umsetzt. In diesem Fall können mehrere Attribute des Programms geändert werden. Am Beispiel eines Terminalprogramms kann dies bedeuten, daß mit Hilfe der Menuefunktionen die Baudrate, die Datenwordbreite und die Anzahl der Stopbits eingestellt werden kann. Das Setzen einer neuen Baudrate würde dann die zuvor gewählte Rate zurücksetzen, während die Attribute für Stopbit und Datenwordbreite unverändert bleiben. Weiterhin erlaubt Intuition dem Programmierer, Menuepunkte (auch als Menueitem oder nur Item bezeichnet), die sich gegenseitig ausschließen, mit einem sogenannten 'Merker' zu versehen. In diesem Zusammenhang werden die Flags CHECKED und CHECKIT verwendet. Sie müssen in der Flag-Variablen der MenuItem-Struktur gesetzt werden, dazu jedoch später mehr. Als Merker stellt Intuition einen Haken zur Verfügung, dieser muß aber nicht benutzt werden, sondern kann auch andere Formen aufweisen. Bild 2 zeigt die Benutzung eines Merkers.

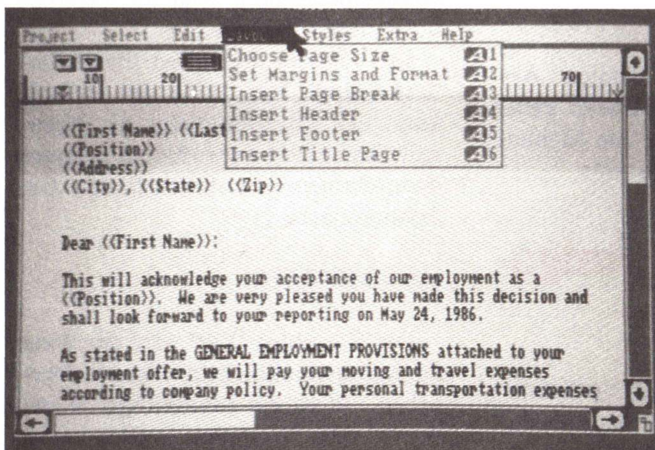


Bild 1: Das Textverarbeitungsprogramm Textcraft benutzt Pull-Down-Menues.

Die Nachrichtenübermittlung

Woher soll ein Programm nun wissen, welches Menue bzw. welcher Menuepunkt vom Benutzer angewählt wurde? Jedes Menue oder jeder Menuepunkt besitzt einen bestimmten Wert. Wählt der Benutzer einen Menuepunkt an, gibt das System einen Wert zurück, der es erlaubt, das angewählte Feld genau zu bestimmen. Bild 3 zeigt an einem kleinen Beispiel auf, wie das System anhand einer zurückgegebenen Zahl den Menuepunkt selektiert.

Um die Zahl in einem Programm auszuwerten, sind drei Funktionen vom Amiga-Betriebssystem bereitgestellt – MENUNUM(Nummer), ITEMNUM(Nummer) und SUBNUM (Nummer).

Beispiel:

Wurde ein Menue angewählt (die rechte Maustaste betätigt ohne das Menuefeld zu benutzen), liefert Intuition den Wert NENUNULL als Menuenummer zurück. Entsprechend gilt dies für die oben aufgeführten Funktionen NOMENU, NOITEM und NOSUB. Ein Vergleich mit diesen vordefinierten Konstanten gibt nun darüber Auskunft, daß eine entsprechende Wahl durch den Benutzer nicht stattgefunden hat.

Das alles hört sich ein wenig kompliziert an, ist in Wirklichkeit aber gar nicht so schwierig, wie es viel-

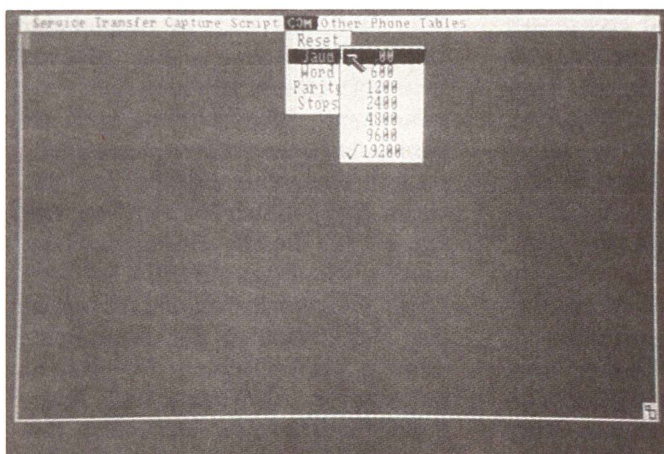


Bild 2:
Die Baudrate wird im abgebildeten Unter--menue mit Hilfe eines Merkers gekennzeichnet.

leicht den Anschein hat. Wenn Sie mein Beispielprogramm am Ende des Artikels aufmerksam studieren, werden Sie die Programmierung leicht nachvollziehen können.

```
Das Programm empfängt vom System folgende Zahl: 1A2C
Dies entspricht Binär:
```

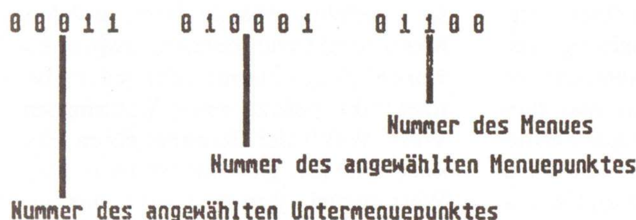


Bild 3:
Das Selektieren eines Menues bzw. Menuepunktes geschieht nach dem abgebildeten Schema.

Menue-Programmierung in C oder BASIC

Wieder stellt sich die Frage nach der Programmiersprache. Jede Sprache kann natürlich herangezogen werden – aber bei welcher hält sich der Programmieraufwand in Grenzen? In der Programmiersprache C lassen sich Menues einfach und sehr komfortabel einrichten. Der Programmieraufwand ist zwar etwas komplex, bietet dafür aber eine Menge verschiedener Möglichkeiten.

Das AmigaBASIC besitzt glücklicherweise Befehle, die das Erstellen von Menues sehr einfach gestalten. So ist BASIC zum Programmieren von Menues zu empfehlen. Andere Programmiersprachen bieten nicht den Luxus von BASIC an. In PASCAL ist das Erstellen von Menues nicht so komfortabel, ebenso in Assembler. Deshalb möchte ich nur auf die Programmierung von Menues in den Programmiersprachen C eingehen sowie auf den mitgelieferten BASIC-Interpreter des Amiga. Andere BASIC-Dialekte besitzen keine Befehle zur Einbindung von Menues oder nur sehr unkomfortable.

Menues in Amiga-BASIC

Beginnen wir mit der Programmierung von Menues in AmigaBASIC. Die Entwickler des Interpreters haben zum Erstellen und Handhaben von Menues einige Befehle entwickelt. Zum einen der 'MENU'-Befehl, der es erlaubt, Menues zu erzeugen, sowie der dazugehörige 'MENU RESET'-Befehl, der das Menue wieder ausschaltet. Die Syntax hierzu lautet:

```
MENU Kennung,Punkt,Status[,Titel]
MENU RESET
```

Die einzelnen Variablen des Befehls haben dabei folgende Bedeutung:

'Kennung' legt die Nummer des Menues fest, der Wert muß zwischen 1 und 10 gewählt werden. 'Punkt' definiert die Nummer des Menuepunktes, die Werte 0 bis 19 können gewählt werden. 'Status' bestimmt den Zustand des Menues bzw. eines Menuepunktes, je nachdem, welcher Wert eingesetzt wird. Dem Anwender stehen hierbei drei Werte zur Verfügung: 0, 1 oder 2. '0' inaktiviert das Menue oder ein Menuepunkt, hierbei kommt es auf den Wert an, der in der Variablen 'Punkt' gesetzt ist. '1' aktiviert das Menue oder den Menuepunkt, und '2' versieht einen Menuepunkt mit einem Merker. Vor dem Titel sollten in diesem Falle zwei Leerzeichen eingefügt werden. 'Titel' bestimmt den Namen des Menues bzw. des Menuepunktes.

Der nächste Befehl ist enorm wichtig, er veranlaßt einen Sprung in eine angegebene Zeile, sobald ein Menue angewählt wurde. Die Syntax des Befehls:

```
ON MENU GOSUB Marke
```

'Marke' ist hierbei der Label, zu dem gesprungen werden soll. Mein Beispielprogramm zeigt die Verwendung dieses Befehls.

Ein weiterer Befehl ist die Menu-Anweisung. Mit Hilfe des Befehls kann der Programmierer die Nummer des zuletzt angewählten Menues oder Menuepunktes herausfinden. Die Syntax hierzu lautet:

```
Var = MENU(n)
```

Der Variablen 'Var' wird der Wert des Menuepunktes oder Menues übergeben. 'n' ist ein ganzzahliger Wert, der '0' oder '1' gesetzt werden kann. '0' gibt nur den Wert eines Menues zurück, und '1' nur den Wert eines Menuepunktes.

Die letzten Befehle, welche die Entwickler dem Interpreter mit auf den Weg gegeben haben, sind die MENU ON/OFF/STOP-Anweisungen. Mit deren Hilfe läßt sich herausfinden, ob ein Menue angewählt wird, um dann die entsprechende Reaktion zu veranlassen. Die Syntaxen der Anweisungen sind wie folgt festgelegt:

MENU ON
MENU OFF
MENU STOP

Die erste Anweisung muß unbedingt aufgerufen werden, um eine Unterbrechungsverzweigung zu bewirken. Nach 'MENU OFF' kann eine solche Verzweigung nicht mehr vollzogen werden. Bei dem Befehl 'MENU STOP' merkt sich AmigaBASIC, daß ein Menue aufgerufen wurde.

Listing 1 am Ende des Artikels zeigt Ihnen, wie das Erstellen und die Abfrage von Menues vonstatten gehen kann. Auf eine nähere Erklärung möchte ich verzichten, da das Listing mit ausreichenden Erläuterungen der einzelnen Programmteile versehen ist. Dieses Programm kann jeder nach Starten des AmigaBASIC-Interpreters eingeben. Bild 4 zeigt, was das kleine Listing bewirkt.

Zu Beginn möchte ich etwas Allgemeinwissen über C verbreiten. Man sollte wissen, daß C eine Modulsprache ist. Man kann C so programmieren, daß ganze Programmme aus einzelnen Modulen bestehen. Das Herzstück eines C-Programms ist immer das 'main()'-Modul. Von diesem Modul werden dann eventuell andere Module oder Unterprogramme angesprungen. Der Umfang der Module wird immer von einer offenen, geschweiften Klammer ({) und einer schließenden, geschweiften Klammer (}) bestimmt. Weiterhin enthält ein C-Programm meistens einen sogenannten Programmkopf, in dem Dateien eingebunden, Variablen definiert oder allgemeine Zuweisungen vorgenommen werden.

Doch nun zur Programmierung von Menues. Menues werden immer in der Titelzeile des jeweiligen Bildschirms ausgegeben, obwohl sie eigentlich mit einem Fenster verkettet werden, das dann wiederum mit dem Bildschirm verbunden ist. Um ein Menue zu erzeugen, müssen folgende Schritte unbedingt erfolgen. Es müssen den Variablen der Menue-Struktur Werte zugewiesen werden. Solche Strukturen sind Sammlungen von Variablen, die das Aussehen und die Möglichkeiten eines Menus festlegen. Es sind zwei Strukturen zum Erstellen von Pull-Down Menues nötig. Zum einen die eigentliche Menue-Struktur, zum anderen eine Struktur, welche die einzelnen Menuepunkte und Untermenuepunkte erstellt. Der Name der letzteren lautet MenuItem-Struktur. Die zwei Strukturen sind in nebenstendem Kasten abgebildet. Sie enthalten alle wichtigen Werte, die von einem Pull-Down Menue unbedingt gebraucht werden.

Die Verwendung der Strukturen und die Zuweisungen der Variablen habe ich in Listing 2 nachvollzogen. Eine Erklärung dieses Listings finden Sie am Ende des Artikels. Doch zuvor möchte ich noch auf Intuition-Funktionen eingehen, die sich speziell mit Menues befassen.

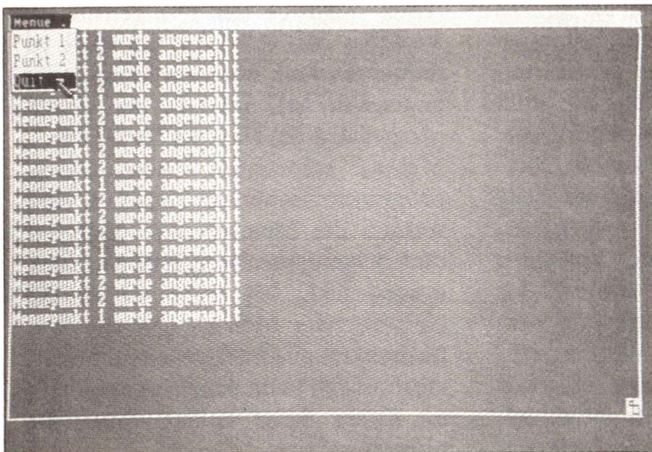


Bild 4: Das abgedruckte BASIC-Programm bewirkt die abgebildete Ausgabe.

Menues in C

Vielleicht ist Ihnen aufgefallen, daß der AmigaBASIC-Interpreter es nicht erlaubt, Submenues (Untermenues) zu erstellen oder Tastatursequenzen den einzelnen Menuepunkten zuzuweisen. In C kann dies unkompliziert geschehen – durchaus ein Plus dieser Sprache. Die Programmierung ist weitaus komplexer, aber auch flexibler als BASIC. Ein C-Neuling wird die Programmierung mit etwas Einarbeitung verstehen und sollte deshalb nicht gleich das Handtuch werfen, wenn er nur C hört.

Programmierung in C

Menue-Funktionen

Am wichtigsten in Bezug auf Menues sind die zwei Funktionen, die es ermöglichen, ein Menue mit einem Fenster zu verbinden oder wieder zu entfernen. Das bedeutet, daß der Menuebalken im Bildschirmbalken installiert bzw. wieder entfernt wird. Die beiden Funktionen besitzen nachstehende Syntax:

```
SetMenuStrip(Window,Menu)
ClearMenuStrip(Window)
```

Die Variable 'Window' stellt in beiden Funktionen einen Zeiger auf

Die Menue Struktur:

```
struct Menu
{
 struct Menu *NextMenu;
 SHORT LeftEdge, TopEdge;
 SHORT Width, Height;
 USHORT Flags;
 BYTE *Menuname;
 struct MenuItem *FirstItem;
};
```

Die MenuItem Struktur:

```
struct MenuItem
{
 struct MenuItem *NextItem;
 SHORT LeftEge, TopEdge;
 SHORT Width, Height;
 USHORT Flags;
 LONG Auswahl;
 APTR ItemFill;
 BYTE Command;
 struct MenuItem *SubItem;
 USHORT NextSelect;
};
```

Die Struktur für ein Sub-Item entspricht der MenuItem-Struktur.

das betreffende Fenster dar. 'Menu' ist ein Zeiger auf das erste Menue in der Menueleiste. Ohne die erste Funktion ist das Ausgeben von Menues nicht möglich.

Um Menues oder Menuepunkte einzeln zu verwalten, stehen einem Programmierer die Funktionen OnMenu(...) und OffMenu(...) zur Verfügung. Um ein ganzes Menue zu desaktivieren bzw. zu aktivieren, setzen Sie den Teil der Menuenummer, der dem zu bearbeiteten Menue entspricht, auf den entsprechenden Wert. Der Teil der Menuenummer, der dem Menuepunkt entspricht, ist auf NOITEM zu setzen. Um einen einzelnen Menuepunkt zu desaktivieren bzw. zu aktivieren, ist die Nummer des Menues, welches den Menuepunkt enthält, zu setzen und die Menuepunktnummer dementsprechend gleichzusetzen. Der Wert für.den Submenuepunkt Teil ist auf NOSUB zu setzen. Um einen einzelnen Submenuepunkt zu beinflussen, ist die Menuenummer entsprechend dem gesamten Zugriffspfad zu setzen. Die Syntax zum Aktivieren lautet:

OnMenu(Window, Menuenummer)

Zum Ausschalten:

OffMenu(Window, Menuenummer)

Die letzte Funktion, die ich behandeln möchte, gestattet es, die Adresse eines Menuepunktes zu ermitteln. Die Syntax dieser Intuition-Funktion ist wie folgt festgelegt:

ItemAdress(MenuStrip, Menuenummer)

Die Variable 'Menustrip' enthält hierbei einen Zeiger auf das erste Menue der verketteten Liste, 'Menuenummer' entsprechend dem Menuepunkt.

Programmerläuterung

Wie immer, habe ich auch zu diesem Kurs ein Beispielprogramm in C geschrieben. Ein C-Compiler muß deshalb in Ihrer Programmsammlung vorhanden sein. Das Programm erstellt zwei Menues, wobei das erste die unterschiedlichen Anwählarten eines Menuepunktes, das zweite den Gebrauch eines Merkers beschreibt. Die Zeilennummern dienen nur zur besseren Orientierung und dürfen auf keinen Fall mit eingegeben werden.

Doch beginnen wir mit der Programmerläuterung. In Zeile 16 und 17 werden zwei Dateien eingebunden. Die Datei Namens 'types.h' enthält Variablen-Definitionen wie 'ULONG' oder 'UBYTE. Die zweite Datei mit dem Namen 'intuition.h' ist schon von größerer Reichweite. Sie enthält nämlich die einzelnen Struktur-Definitionen, unter anderem auch die zur Menueprogrammierung. Außerdem werden in dieser Datei die verschiedenen Flagwerte gesetzt. Die beiden Zeilen 23 und 24 bekommen verschiedene Strukturen-Zeiger zugewiesen. Dies sind die IntuitionBase und die IntuiMessage Struktur. Die Zeiger werden später im Hauptprogramm benötigt. Weiter geht es mit der Variablen-Definition. Sie muß nicht unbedingt vorgenommen werden, dient aber der besseren Übersichtlichkeit. Als nächstes, ab Zeile 43, folgt die NewWindow-Struktur. Hier werden den Fenster-Variablen Werte zugewiesen. Von Zeile 63 bis 131 erstrecken sich IntuiText-Strukturen. Diese Strukturen dienen dazu, einem Text ein bestimmtes Aussehen zu geben. Sie werden für die einzelnen Texte der Menues und Menuepunkte benötigt. Ab Zeile 138 bis zur Zeile 214 werden die einzelnen Menuepunkte definiert, hierzu dienen, wie Sie wissen, die MenuItem-Strukturen. Danach folgen die Strukturen der Menues. Mit den Menu Strukturen ist der Programmkopf abgeschlossen und das Hauptprogramm beginnt in der Zeile 244. Zwei Variablen werden zunächst als ULONG- und USHORT-Wert deklariert, bevor zwei zusätzliche Zeiger auf Strukturen definiert werden. Danach wird die Intuition-Library geöffnet. Man muß dazu wissen, daß nur Funktionen oder Routinen einer Library aufgerufen werden können, wenn diese zuvor geöffnet wurde. Nicht anders verhält es sich in unserem Beispiel. Das Öffnen wurde so programmiert, daß bei einem mißlungenen Versuch das Programm seinen Ablauf beendet. Das Öffnen des Fensters geschieht von den Zeilen 266 bis 268. Konnte das Fenster aus irgendeinem Grunde nicht geöffnet werden, beendet auch hier das Programm seinen Ablauf. In Zeile 274 tritt die Funktion 'SetMenuStrip()' in Aktion. Sie verkettet den globalen Menue-Aufbau (Menuetext, Menueaufbau und Menuepunkte) mit dem Fenster. Ab Zeilennummer 278 beginnt die Hauptschleife des Programms, der Ausdruck 'for(;;)' stellt hierbei eine Endlosschleife dar. Die 'if'-Anweisung, ab Zeile 283, fragt bestimmte Ereignisse ab. Unter anderem, ob das Closewindow-Gadget mit der Maus angewählt wurde, oder ein Menu. Beim Anwählen des Gadgets wird das Programm beendet und alle Einrichtungen werden geschlossen. Beim Anwählen eines Menues jedoch wird zunächst eine weitere 'if'-Abfrage durchgeführt. Wenn der gelieferte 'code' ungleich der fest definierten Variable MENUNULL ist (dies ist immer der Fall, wenn ein Menupunkt angewählt ist), wird in ein Unterprogrramm des Namens 'Bearbeite__Menu()' gesprungen. Dieses Unterprogramm beginnt ab Zeile 314. Hier werden die Menuepunkte selektiert und dementsprechend eine Meldung ausgedruckt. Ich habe jeweils die Nummer des Menues und des Menuepunktes ausdrucken lassen. Selbstverständlich kann der Programmierer an dieser Stelle andere Ereignisse programmieren.

Wie geht's weiter?

Nachdem nun Bildschirme, Fenster, Gadgets und Menues behandelt wurden, möchte ich im nächsten Intuition-Kursteil die Programmierung von System-Requestern und 'normalen' Requestern aufgreifen. Auch dieses Kapitel wird Sie sicher interessieren. Es würde mich freuen, wenn Sie dann auch wieder dabei sind.

(AK)

Tabellarische Übersicht

Die Erklärungen der Variablen in der Menue-Struktur. Durch die Festlegung dieser Variablen kann ein Menue definiert werden.

Die Bedeutungen der einzelnen Felder sind:

NextMenue
Zeiger auf die nächste Menue-Struktur in der Liste. Das letzte Element enthält den Wert NULL.

LeftEdge, TopEdge
LeftEdge enthält den Offset vom linken Bildschirmrand, an dem die Variable Menuename ausgegeben werden soll. TopEdge wird in der jetzigen Version nicht verwendet. Dieser Wert wird auf Null gesetzt.

Width, Height
Width gibt die Anzahl von Pixeln an, die für die Anzeige von Menuenamen reserviert werden soll (dieser Wert soll nicht kleiner sein als die für Menuename wirklich benötigten Pixel). Height wird in dieser Version nicht verwendet.

Flags
Die zulässigen Flags sind:

MenueNABLED
Dieses Flag gibt an, ob das Menue für den Benutzer anwählbar ist oder nicht. Es kann nachträglich mit Hilfe der Funktionen OnMenue() und OffMenue() verändert werden.

MIDRAWN
Das Flag kontrolliert, ob die einzelnen Menuepunkte angezeigt werden oder nicht.

MenueName
Enthält einen Zeiger auf den Menuenamen, der in der Menue Zeile angezeigt werden soll. Der String muß mit ASCII Null enden.

FirstItem
Enthält einen Zeiger auf die erste MenueItem-Struktur, die mit dem Menue verkettet werden soll.

Die Variablen der MenueItem Struktur haben folgende Bedeutung:

NextItem
Die Variable enthält eine Zeiger auf die nächste MenueItem Struktur. Die letzte Struktur in der Liste enthält hier den Wert NULL.

LeftEdge, TopEdge
Diese Variablen enthalten die Positionen des Feldes, innerhalb dessen die Menue-Funktion angewählt werden kann. Left-Edge ist relativ zu der in der Menue-Struktur gesetzten Left-Edge-Variable. TopEdge ist relativ zu der von Intuition gesetzten Top-Edge-Variable. Der Wert Null positioniert das Feld an der höchst möglichen Stelle am Bildschirm (in Abhängigkeit der Bildschirmauflösung, des verwendeten Zeichensatzes usw).

Width, Height
Die Variablen beschreiben die Breite bzw. die Höhe des Menuefeldes.

Flags
Die nachstehenden Flags können in dieser Variable gesetzt werden:

CHECKIT
Das Flag kennzeichnet diesen Menuepunkt als Programmattribut und erlaubt die Verwendung des Merkers und der Variable Auswahl.

CHECKED
Dieses Flag zeigt an, ob das Attribut, welches durch diesen Menuepunkt repräsentiert wird, mit einem Merker gekennzeichnet ist oder nicht. Es wird entsprechend der Variable Auswahl gesetzt.

ITEMTEXT
Das Flag kennzeichnet den Zeiger ItemFill als einen Zeiger auf eine IntuiText-Struktur. Soll der Menuepunkt eine Graphik enthalten, so darf diese Flag nicht gesetzt werden.

COMMSEQ
Mit Hilfe dieses Flags wird gekennzeichnet, daß die Variable Command einen gültigen Wert enthält, der als alternative Tastensequenz verwendet werden soll.

ITEMENABLED
Dieses Flag bestimmt, ob das dazugehörige Item durch den Benutzer anwählbar ist oder ob es zur Zeit nicht verwendbar ist (schwach gezeichnet).

HIGHFLAGS
Der Zustand, in dem gekennzeichnet werden soll, daß das Item angewählt ist, kann durch folgende Werte bestimmt werden.

HIGHCOMP
Dieser Wert veranlaßt Intuition, alle Pixels zu komplementieren.

HIGHBOX
Das Flag bewirkt die Kennzeichnung durch einen viereckigen Rand um das Item.

HIGHIMAGE
Diese Flag bringt die durch die Variable SelectFill bestimmte Struktur anstelle der durch die ItemFill Variable gesetzte Struktur zur Anzeige. HIGHNONE Dieses Flag wird gesetzt, wenn keine Veränderung der Anzeige bei Anwahl erfolgen soll.

ISDRAWN
Dieses Flag wird von Intuition gesetzt, wenn die zu dem Item gehörenden Sub-Items dem Benutzer angezeigt werden.

HIGHITEM
Das Flag wird von Intuition gesetzt, wenn der Zeiger (Maus) sich momentan über dem Feld des Items befindet.

Auswahl
Eine 16-Bit Zahl entsprechend den von Ihnen gewümschten Funktionen auf das Flag CHECKED.

ItemFill
Ein Zeiger auf die Struktur, die anzeigt, wenn das Item auf dem Bildschirm dargestellt wird.

SelectFill
Diese Variable zeigt bei gesetztem HIGHIMAGE-Flag auf eine alternative Struktur, die anstelle der durch die Item-Fill-Variable bestimmte angezeigt werden soll.

Command
Die Variable enthält ein ASCII-Zeichen, das zusammen mit der rechten Amiga-Taste die alternative Kommandosequence bestimmt.

SubItem
Ein Zeiger auf eine Sub-Item-Struk-

tur. Befinden Sie sich bereits in einer solchen, wird dieser Wert ignoriert.

NextSelect
Diese Variable wird von Intuition ungleich MenueNULL gesetzt, wenn im internen Eingabepuffer von Intuition keine weitere Menueanwahl auf ihre Ausführung wartet.

```
/*********************************************************
*                                                       *
* Erstellung verschiedener Menues                       *
* Autor: Andreas Krämer                                 *
* Datum: 7.August.1987                                  *
* Copyright by KICKSTART 1987                           *
* Erstellt mit dem Lattice C Compiler                   *
*                                                       *
*********************************************************/

/*********************************************************
* Zwei 'include' Dateien werden in das Programm         *
* eingebunden.                                          *
*********************************************************/

#include <exec/types.h>
#include <intuition/intuition.h>

/* Zuweisungen von verschiedenen Strukturen mit einem   *
*  Zeiger. Die Zeiger werden später im Hauptprogramm    *
*  benötigt.                                           */

struct IntuitionBase *IntuitionBase;
struct IntuiMessage *Nachricht;

/* Variablen Definition */

#define INTUITION_REV   29
#define Blau            0
#define Weiss           1
#define Schwarz         2
#define Rot             3
#define Million     1000000
#define Menu_1          0
#define Menu_2          1
#define Menu_3          2
#define Item_1          0
#define Item_2          1
#define Item_3          2

/* Die NewWindow Struktur */

struct NewWindow NeuesFenster =
{
 20, 20, 400, 100,
 Rot, Weiss,
 CLOSEWINDOW | MENUPICK,
 WINDOWDRAG | WINDOWSIZING | ACTIVATE
 | SIZEBBOTTOM | SIZEBRIGHT | WINDOWCLOSE,
 NULL, NULL,
 "Mein Fenster",
 NULL, NULL,
 10, 10, 640, 200,
 WBENCHSCREEN
};

/*********************************************************
* Die IntuiText Struktur wird benötigt um den Menue     *
* Text zu erstellen. Ein Zeiger der MenuItem Struktur   *
* weist auf die jeweilige IntuiText Struktur.           *
*********************************************************/

struct IntuiText MenuText1a =
{
   Rot,Weiss, /* Farbe des Textes */
   JAM2,      /* Ausgabe Modus    */
   5,5,       /* Position des Textes */
   NULL,      /* Kein spezieller Zeichensatz */
   "Erster Versuch", /* Menue Text */
   NULL       /* Kein Zeiger auf eine weitere */
};            /* IntuiText Struktur          */

struct IntuiText MenuText1 =
{
   Rot,Weiss,
   JAM2,
   5,5,
   NULL,
   "Angewählt     ",
   NULL
};

struct IntuiText MenuText2 =
{
   Rot,Weiss,
   JAM2,
   5,4,
   NULL,
   "Zweiter Versuch",
   NULL
};

struct IntuiText MenuText2a =
{
   Rot,Weiss,
   JAM2,
   5,5,
   NULL,
   "Dritter Versuch",
   NULL
};

struct IntuiText MenuText3 =
{
   Rot,Weiss,
   JAM1,
   CHECKWIDTH,2,
   NULL,
   "hallo1",
   NULL
};

struct IntuiText MenuText4 =
{
   Rot,Weiss,
   JAM1,
   CHECKWIDTH,2,
   NULL,
   "hallo2",
   NULL
};

struct IntuiText MenuText5 =
{
   Rot,Weiss,
   JAM1,
   CHECKWIDTH,2,
   NULL,
   "hallo3",
   NULL
};

/**************************************************
* Die MenuItem Struktur. Die Definition der      *
* einzelnen Menuepunkte.                         *
**************************************************/

struct MenuItem DrittesMenu_Item =
{
   NULL,
   10,40,130,16,
   ITEMTEXT | ITEMENABLED | HIGHCOMP,
   0,
   (APTR)&MenuText2a,
   NULL,
   0,
   NULL,
   0
};

struct MenuItem ZweitesMenu_Item =
{
   &DrittesMenu_Item,
   10,20,130,16,
   ITEMTEXT | ITEMENABLED | HIGHBOX,
   0,
   (APTR)&MenuText2,
   NULL,
   0,
   NULL,
   0
};

struct MenuItem ErstesMenu_Item =
{
   &ZweitesMenu_Item,
   10,0,135,16,
   ITEMTEXT | ITEMENABLED | HIGHIMAGE,
   0,
   (APTR)&MenuText1a,
   (APTR)&MenuText1,
   0,
   NULL,
   0
};

struct MenuItem Menu_Item_Drei =
{
   NULL,
   30,40,100,15,
   ITEMTEXT | ITEMENABLED | HIGHCOMP | CHECKIT ,
   0xFFFB,
   (APTR)&MenuText5,
   NULL,
   0,
   NULL,
   0
};

struct MenuItem Menu_Item_Zwei =
{
   &Menu_Item_Drei,
   30,20,100,15,
   ITEMTEXT | ITEMENABLED | HIGHCOMP | CHECKIT ,
   0xFFFD,
   (APTR)&MenuText4,
   NULL,
   0,
   NULL,
   0
};

struct MenuItem Menu_Item_Eins =
{
   &Menu_Item_Zwei,
   30,0,100,15,
   ITEMTEXT | ITEMENABLED | HIGHCOMP | CHECKIT | CHECKED,
   0xFFFE,
   (APTR)&MenuText3,
   NULL,
   0,
   NULL,
```

```
   0
};

/**********************************************************
* Die Menue Strukturen.                                   *
**********************************************************/

struct Menu  ZweitesMenu =
{
   NULL,
   150,0,150,0,
   MENUENABLED,
   " EIN ZWEITES ",
   &Menu_Item_Eins
};

struct Menu  ErstesMenu =
{
   &ZweitesMenu,
   0,0,150,0,
   MENUENABLED,
   " EIN ERSTES MENUE ",
   &ErstesMenu_Item
};

/*******************************************************
*                                                      *
* Beginn des Hauptprogrammes                           *
*                                                      *
*******************************************************/

main()
{

/* Zuweisung von zwei Variablen als ULONG und USHORT   *
*  Wert. Werden später zur Ereignisabfrage benötigt.  */

ULONG   NachrichtenArt;
USHORT  code;

/* Zusätzliche Zeiger auf Strukturen */

struct Window *Window;
struct Message *GetMsg();

/* Öffnen der Intuition Library */

IntuitionBase = (struct IntuitionBase *)
      OpenLibrary("intuition.library",INTUITION_REV);
if(IntuitionBase==NULL)exit(FALSE);

/* Öffnen des Fensters */

if((Window = (struct Window *)
      OpenWindow(&NeuesFenster))==NULL)
exit(FALSE);

/*************************************************
* Das Verketten des Menues mit dem Fenster.      *
*************************************************/

SetMenuStrip(Window,&ErstesMenu);

/* Die Hauptschleife des Programms */

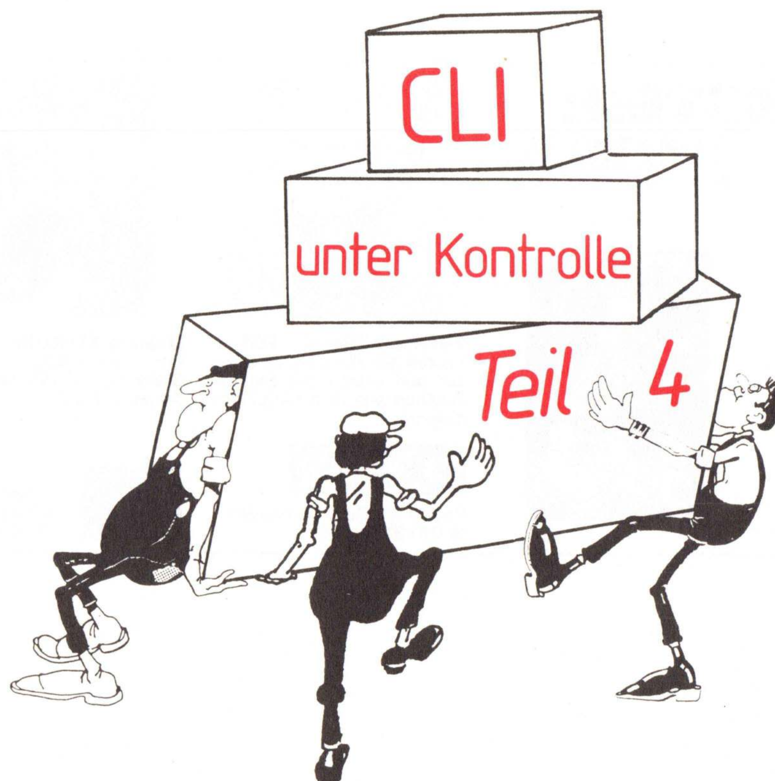
for(;;)
  {
   /*******************************************************
   * Die Schleife fragt bestimmte Ereignisse ab.          *
   *******************************************************/
    if(Nachricht=(struct IntuiMessage *)
                 GetMsg(Window->UserPort))
     {
       NachrichtenArt=Nachricht->Class;
       code=Nachricht->Code;
       ReplyMsg(Nachricht);
       switch(NachrichtenArt)
         {
           /* Abfrage des CLOSEWINDOW Gadgets */
           case CLOSEWINDOW : ClearMenuStrip(Window);
                              CloseWindow(Window);
                              CloseLibrary(IntuitionBase);
                              exit(TRUE);
                              break;

           /* Die Menueabfrage */
           case MENUPICK    : if (MENUNUM(code) != MENUNULL)
                                 Bearbeite_Menu( MENUNUM(code),
                                                 ITEMNUM(code),
                                                 SUBNUM(code));
                              break;
         } /* Ende switch */
     } /* Ende if */
  } /* Ende for */
return(0);
} /* Ende des Hauptprogrammes */

/**********************************************************
* Das Unterprogramm zu den einzelnen Menuepunkten         *
**********************************************************/

Bearbeite_Menu(menu,item,subitem)
int menu,item,subitem;
{
  switch(menu)
    {
     /* Erste Menueabfrage */
      case Menu_1 : switch(item)
                      {
                        case Item_1 : printf("menu1 item1\n");
                                      break;
                        case Item_2 : printf("menu1 item2\n");
                                      break;
                        case Item_3 : printf("menu1 item3\n");
                                      break;
                      }
                    break;

     /* Zweite Menueabfrage */
      case Menu_2 : switch(item)
                      {
                        case Item_1 : printf("menu2 item1\n");
                                      break;
                        case Item_2 : printf("menu2 item2\n");
                                      break;
                        case Item_3 : printf("menu2 item3\n");
                                      break;
                      }
                    break;
    }
return(0);
}
```



Das CLI und seine Tücken

Immer weiter geht es mit dem Command-Line-Interpreter. Noch sind nicht alle Hürden genommen. Aber Schritt für Schritt werden die Unklarheiten beseitigt. Rom ist auch nicht an einem Tag erbaut worden, genauso verhält es sich mit dem CLI. Zum Profi in dieser Oberfläche zu werden, ist ein langer Weg.

Weitere CLI-Befehle

Im vierten Teil des CLI-Kurses möchte ich folgende Befehle aufgreifen: 'makedir', 'delete', 'rename' und 'protect'. Diese Befehle werden auch als Datei-Befehle bezeichnet, weil sie sich mit der Verwaltung und Handhabung von Dateien befassen. Einige dieser Datei-Befehle haten wir schon früher besprochen. Dazu gehörten 'dir', 'list' und 'copy'.

Die nun folgenden Befehle hängen zum Teil eng miteinander zusammen. Beginnen wir zunächst mit dem Befehl 'makedir'.

'makedir'

Der 'makedir'-Befehl ist dazu gedacht, neue Verzeichnisse oder Directories zu erstellen. Dabei spielt es keine Rolle, in welcher Ebene Sie sich befinden. Das noch so 'tiefe' Unterverzeichnis erlaubt die Durchführung des Befehls. Mit seiner Hilfe können Sie bequem Ordnung auf ihrer Diskette schaffen. Die Syntax des Befehls ist recht einfach:

makedir Verzeichnis

'Verzeichnis' stellt hierbei den Namen des neuen Directories dar. Der Ausdruck

makedir Spiele

legt beispielsweise ein Verzeichnis mit dem Namen 'Spiele' an. Selbstverständlich erlaubt es der Command-Line-Interpreter auch, Pfade einzugeben. Hierzu folgendes Beispiel:

makedir df1:Adressen/Bekannte

Es wird ein Unterverzeichnis in Laufwerk 'df1' im Verzeichnis Adressen erstellt. Es ist jedoch unbedingt zu beachten, daß das Verzeichnis Adressen bereits existiert. Ebenfalls ist es möglich, Verzeichnisse mit demselben Namen zu erstellen. Lediglich im aktuellen Verzeichnis ist dies nicht möglich. Das System gibt dann aber eine Fehlermeldung aus.

Nicht nur einfach 'delete'

Der 'delete'-Befehl gestattet es einem Anwender, Dateien oder Verzeichnisse zu löschen. Damit ist aber nicht zu spaßen, denn schnell sind Programme gelöscht, die man doch noch gebrauchen kann. Auch ein Tippfehler kann fatale Folgen haben. Vorsicht ist oberstes Gebot. Man sollte die eingegene Zeile immer überprüfen, bevor man die 'Return'-Taste betätigt. Zur Syntax:

delete Name[Name...][all][qlquiet]

'Name' beschreibt den Namen der Datei oder des Verzeichnisses, das gelöscht werden soll. Ein Verzeichnis kann allerdings nur gelöscht werden, wenn sich keine Datei oder Verzeichnis mehr darin befindet. 'all' veranlaßt in Zusammenhang mit dem 'delete'-Befehl alle Dateien, auf die der Name paßt, zu löschen. Da es erlaubt ist, sogenannte Joker zu verwenden, ist es ohne weiteres möglich, sehr viele Dateien auf einmal zu löschen.

'q' oder 'quiet' unterdrückt die normalerweise angezeigte Löschbestätigung.

Ich möchte noch einige Beispiele des 'delete'-Befehls zeigen. Tippen Sie die Beispiele allerdings unter keinen Umständen ab, da dies katastrophale Folgen haben könnte.

delete Adressen

löscht die Datei 'Adressen'

delete Adressen/Bekannte1 Platten/Pop cds

löscht zunächst die Datei 'Bekannte', danach das Verzeichnis Adressen, dann die Datei 'Pop' und das Verzeichnis 'Platten' und schließlich noch die Datei 'cds'. Sie sehen, Sie können mehrere Dateien mit einem 'delete' löschen, ohne Joker oder Wildcards zu verwenden. Nach einem Tippfehler werden jedoch die danach folgenden Dateien nicht mehr gelöscht, weil das System die Fehlermeldung ausgibt, daß die Datei nicht gefunden werden konnte. Die ganze Schreibarbeit war umsonst. Wollen Sie nur Dateien mit derselben Extension löschen, können Sie 'delete' folgendermaßen anwenden:

delete #?.doc

Es werden nur Dateien gelöscht, die die Extension '.doc' besitzen. Sie sehen, hier wurden die Joker '#?' verwendet.

delete df1:#?.(doc/txt) all

Dieses Beispiel hat gleich zwei Neuerungen. Zum einen wurde hier 'all' mit einbezogen, zum anderen der Ausdruck '(doc/txt)'. Zur Erklärung: Im Laufwerk 'df1' sollen alle Dateien, die die Extension 'doc' oder

'txt' besitzen, gelöscht werden, und zwar auch in eventuell vorhandenen Unterverzeichnissen. Dies bewirkt der 'all'-Ausdruck.

Noch ein Beispiel:

delete A#? all quiet

löscht auf der ganzen Diskette alle Dateien, die mit 'A' beginnen, also auch die Dateien, welche sich in Unterverzeichnissen befinden. Das System gibt aber keine Löschbestätigung, der Ausdruck 'quiet' ist dafür verantwortlich. Für 'quiet' brauchen Sie auch nur 'q' eingeben, die Abkürzung hat dieselbe Wirkung.

Eine ganz andere Möglichkeit, Dateien zu löschen, besteht mit dem 'dir'-Befehl. Dieser Befehl gestattet es, mit einer bestimmten Option eine Abfrage zu bewirken. Diese Option lautet 'opt i'. Das Beispiel:

dir opt i

veranlaßt den Rechner, nach jedem Anzeigenschritt eine Eingabe zu erwarten. Geben Sie an dieser Stelle den Ausdruck 'del' ein, wird die Datei gelöscht. Der Ausdruck

dir opt ai

zeigt alle Dateien der Diskette an, wobei auch hier eine Eingabe vom System nach jeder Anzeige erfolgen muß. 'del' löscht wiederum die Datei. Die genaue Erklärung des 'dir'-Befehls, mit allen Optionen, war in der KICKSTART-Ausgabe 1 zu lesen.

Noch einmal: Benutzen Sie diesen Befehl nur bei absoluter Sicherheit. Der unsachgemäße Einsatz kann irreparable Folgen nach sich ziehen. Kontrollieren Sie immer die eingegebene Zeile, bevor die 'Return'-Taste gedrückt wird.

Der Schutz vor 'delete'

Die Systemprogrammierer haben jedoch etwas weiter gedacht, um beispielsweise das unbeabsichtige Löschen einer Datei zu vermeiden – der 'protect'-Befehl. Leider mußte ich einige Fehler dieses Befehls und einen Druckfehler feststellen. Doch dazu später mehr. Die Syntax des Befehls ist laut Handbuch folgende:

protect [File]Datei[Flags]Status

'Datei' repräsentiert die Datei, deren Schutzstatus verändert werden soll. 'Status' steht für vier Schlüsselbuchstaben, die angeblich einen bestimmten Schutz bewirken sollen. Bei der Anzeige des momentanen Verzeichnisses mit Hilfe des 'list'-Befehls finden Sie in der dritten Spalte meistens vier Buchstaben: 'rwed'. Diese Buchstaben bestimmen den Schutzstatus des Programms. 'r' legt fest, daß nur lesender Zugriff erlaubt ist, 'w' nur schreibender Zugriff, 'd', daß die Datei gelöscht werden kann und 'e', daß die Datei ausgeführt werden darf. Diese Option soll nur bei ausführbaren Dateien gelten. Soweit, sogut. Der Ausdruck

protect text

müßte, laut Beschreibung des Handbuches, folgendes bewirken: Das Programm 'text' kann weder gelesen, noch beschrieben, gelöscht oder ausgeführt werden. Eine Überprüfung mit dem 'list'-Befehl besagt auch den neuen Schutzstatus. 'text' war bei mir eine reine ASCII-Datei. Nach Eingabe von 'ed text' jedoch ließ sich die Datei problemlos laden und nach einigen Änderungen auch wieder abspeichern. Auch das Starten konnte durchgeführt werden. Lediglich das Löschen von 'text' war nicht durchführbar. 'r','w' und 'e' sind demnach völlig sinnlos. Das Handbuch erklärt hier Dinge, die in der Praxis gar nicht funktionieren. Aber allein die Möglichkeit, ein Programm unlöschbar zu machen, ist schon viel Wert. Damit können wichtige Programme vor unbeabsichtigem Löschen geschützt werden. Das Handbuch der AmigaDOS-Befehle besitzt bei der Erklärung des Befehls 'protect' zu allem Überfluß auch noch einen Fehler. Beispiel 1 müßte, wenn überhaupt, folgendermaßen aussehen:

PROTECT Beschreibung FLAGS R

Natürlich funktioniert dieses Beispiel nicht, ebenso wie das abgedruckte zweite Beispiel. Die einzige funktionierende und sinnvolle Variante des Befehls lautet:

protect text rwe

Nach Eingabe dieser Zeile kann die Datei 'text' nicht mehr gelöscht werden. Der Befehl hat trotz seiner nicht vorhanden Optionen einen nicht zu unterschätzenden Wert.

Der 'rename'-Befehl

Mit diesem Befehl ist es möglich, Dateien oder Verzeichnisse umzubenennen, also einen neuen Namen zuzuordenen. Die Syntax ist wie folgt:

rename [FROM]Adatei[TO|AS]
Ndatei

'Adatei' stellt die umzubenennende Datei dar, 'Ndatei' hierbei den neuen Namen. Mit dem Befehl ist es nicht nur möglich, Umbenennungen von Namen vorzunehmen, sondern auch die Hierachie von Verzeichnissen und Dateien zu ändern. Hierzu ein Beispiel:

rename PC as df0:Expansion/PC

Das Verzeichnis 'PC', das sich zuvor im Hauptverzeichnis, also in der obersten Ebene, befand, wurde in das Verzeichnis 'Expansion' verlagert. Die Hierachie des globalen Dateiverzeichnisses der Diskette hat sich also verschoben. Eine Umbenennung muß nicht vorgenommen werden.

'rename' als 'copy'-Befehl mißbraucht

Mit dem 'rename'-Befehl kann auch innerhalb einer Diskette kopiert werden. Nachstehenden Zeilen bewirken folgendes:

rename clock as :empty/clock rename clock.info as :empty/clock.info

Dieses Beispiel kopiert die Datei 'clock', die sich auf jeder Workbench-Diskette befindet, in den Ordner 'empty'; er befindet sich ebenfalls auf der Workbench. Gleiches gilt für das 'clock.info'. Wie Sie vielleicht wissen, stellt diese Datei das Piktogramm von 'clock' dar. Ohne das Kopieren der '.info' Datei wäre das Piktogramm noch im Hauptdirectory. Nach Anwählen mit der Maus würde aber nichts geschehen.

Wie Sie sehen, haben manche CLI-Befehle ihre Fehler, Stolpersteine oder auch Einsatzmöglichkeiten, über die sich das mitgelieferte Handbuch ausschweigt. Neben den hier behandelten Befehlen 'makedir', 'delete', 'rename' und 'protect' geht es in der nächsten KICKSTART-Ausgabe weiter im CLI-Dschungel. Wenn Sie dann auch wieder dabei sind, ist es nur eine Frage der Zeit, wann Sie zum CLI-Profi werden.

(AK)

C-KURS TEIL 2

Nachdem der erste Teil eine Einführung in den Umgang mit C und dem Compiler war, geht es nun mit der Programmierung von Schleifen weiter. Gerade dieses Gebiet trägt sehr zu der Mächtigkeit einer Sprache bei, ist jedoch für einen Anfänger nicht leicht zu verstehen.

for

Der erste Befehl, der mit Schleifenprogrammierung in Verbindung steht und mit dem wir uns beschäftigen werden, ist der 'for'-Befehl. Manchen wird er schon aus anderen Programmiersprachen bekannt sein. So taucht er z. B. in Basic als

```
10 for i=1 to 10
...
40 next i
```

auf. Auch die weit verbreitete Sprache Pascal hat mit

```
for i:=1 to 10 do ...
```

einen entsprechenden Befehl implementiert. Die C-Variante weist im Gegensatz zu diesen Beispielen einige Besonderheiten auf. Schauen wir uns jedoch zuerst die Syntax des 'for'-Befehls in C an:

```
for( <Befehl> ; <Bedingung> ;
<Befehl>) <Befehl>;
```

Gehen wir die Zeile einmal Schritt für Schritt durch. Das erste <Befehl> enthält die Initialisierung der Schleife; d. h. dieser Befehl wird am Anfang EINMAL ausgeführt. Dies entspricht ungefähr dem Teilstück

```
10 for i=1 ...
bzw.
for i:=1 ...
```

in Basic oder Pascal. Die Initialisierung der 'for'-Schleife in C ist nicht zwingend. Sie kann auch entfallen. Sie können weiter unten ein solches Beispiel finden.

Das nächste Element in der Zeile ist das Semikolon ';'. Es dient zum Trennen der einzelnen Bestandteile innerhalb der Klammern. Das Semikolon wird an dieser Stelle von der Sprachdefinition in C vorgeschrieben. Es darf auch dann nicht entfallen, wenn eines der in den eckigen Klammern ('<','>') aufgeführten Elemente weggelassen wird.

Nun folgt die Bedingung, die zum Abbruch der Schleife führt. Eine 'for'-Schleife wird so oft durchlaufen, wie der Ausdruck der durch <Bedingung> formuliert ist, eine WAHRE Aussage enthält. Der Test auf 'wahre Aussage' erfolgt vor dem Eintritt in die Schleife, d. h. ist die Bedingung vor vorherein falsch, wird die Schleife nie durchlaufen. Man kann diesen Abschnitt der 'for'-Schleife in C mit

```
...to 10 ...
```

in Basic bzw. Pascal vergleichen, obgleich die Konstruktionen in diesen Programmiersprachen starrer sind, als dies bei C der Fall ist. Auch diesen Teil innerhalb des Schleifenaufbaus kann man in C weglassen (Semikolon nicht vergessen). Beachten Sie jedoch, daß eine Schleife ohne Abbruchbedingung endlos abläuft, wenn Sie nicht einen anderen Ausgang eingebaut haben (auf diese Besonderheit komme ich später noch zu sprechen).

Das fast letzte Element einer 'for'-Schleife ist das <Befehl>, das am Ende innerhalb der Klammern steht. Die Anweisung, die an dieser Stelle steht, dient zum Re-Initialisieren der Schleife. Diese bedeutet, daß die Anweisung <Befehl>, die an dieser Stelle steht, nach jedem Schleifendurchlauf ausgeführt wird. Vergleichen wir diese Möglichkeit von C mit Basic oder Pascal:

```
40 next i
bzw.
  ...to 10...
  ...downto -10...
```

Wie Sie sehen, ist C mit seiner 'for'-Schleife doch um einiges flexibler als die Sprachen Basic und Pascal (Pascalfreunde werden zugeben müssen, daß selbst der 'step'-Befehl die Vielseitigkeit der C-Variante nicht erreicht). Auch diesen Teil der 'for'-Schleife kann man weglassen. Nun sind wir beim letzten Teil der Schleifenkonstruktion angekommen. Das hier in den Klammern stehende <Befehl> steht für den Befehl oder eine mit geschweiften Klammern ('{','}') zusammengefaßte Befehlsfolge, die innerhalb eines Schleifenablaufes abgearbeitet werden soll. Dieser Teil entspricht in Basic allen Befehlen, die im oberen kleinen Beispiel zwischen den Zeilen 10 und 40 stehen würden. Die Pascalvariante erinnert hier schon mehr an den Aufbau in C:

```
  ...do <Befehl> ;
bzw.
  ...do begin
        <Befehl>;
          ...
        <Befehl>;
      end;
```

Nach dieser reichlich trockenen Beschreibung einer 'for'-Schleife folgen jetzt endlich die ersten Beispiele für die Praktiker unter Ihnen.

```
main()

    for(;;);
```

Dieses erste Beispiel zeigt die trivialste Form einer 'for'-Schleife in C. Sie enthält weder eine Initialisierung eines Startwertes, noch eine Überprüfung auf Abbruch, sowie eine Re-Initialisierung bzw. einen Befehl, der innerhalb eines Durchlaufes ausgeführt werden soll. Diese Schleife macht also nichts, außer unseren Rechner zu blockieren. Je nach Art des verwendeten Compilers bzw. Rechners (es gibt auch andere als den Amiga!) kann eine solche Schleife nur durch einen Reset des Rechners beendet werden.

Nicht in allen Fällen ist eine Endlosschleife so einfach zu erkennen wie in diesem Beispiel. So passiert es selbst fortgeschrittenen C-Anwendern, daß eine Abbruchbedingung nicht greift. Gerade ein Vergleich auf Gleichheit einer Zahl ist im Gegensatz zu 'größer gleich' bzw. 'kleiner gleich' eine unsichere Sache, die man möglichst vermeiden sollte. Das Beispiel enthält trotz seiner offensichtlichen Unsinnigkeit eine oft gebrauchte Konstruktion in C-Programmen. Es ist dies der Teil

```
for(;;)
```

Man findet diesen Teil häufiger in Programmen, die eine immer wiederkehrende Aufgabe zu erfüllen haben. Nach dem ersten Programmteil mit Variableninitialisierung und ähnlichem wird der Rumpf des Programmes mit for(;;) zusammengefaßt.

```
main()
{
   /* Setzen aller Startwerte */
          ...
   for(;;)
       {
   /* immer wiederholter Teil */
          ...
       }
}
```

Beschäftigen wir uns mit einem Beispiel, das alle Möglichkeiten der 'for'-Schleife in C ausnutzt. Es soll ein Programm entstehen, das die Zahlen Eins bis Zehn auf den Bildschirm ausgibt.

```
main
{
/* Variablendefinition */
   int i;
/* die Schleife */
   for(i=1;i<=10;i++)
      printf("%d\n",i);
/* Programmende */
}
```

Da Sie den Rest des Programms verstehen sollten (Teil 1), schauen wir uns die Schleife einmal genauer an. Als erstes haben wir eine Initialisierung eines Startwertes vorliegen. Dies geschieht durch die Anweisung 'i=1'. Umgangssprachlich könnte man dies so formulieren:

1. Setze die Variable i auf den Wert Eins.

Als zweites wird nun kontrolliert, ob die mit 'i<=10' gegebene Abbruchbedingung erfüllt ist:

2. Wenn die Variable i kleiner oder gleich dem Wert Zehn ist, führe den nächsten Schritt aus. Ansonsten beende die Schleife.

Ist die Abbruchbedingung nicht erfüllt, wird nun der Befehl hinter der schließenden runden Klammer ausgeführt. In unserem Fall 'printf("%d n",i)'.

3. Führe den der Schleife zugeordneten Befehl bzw. Befehlsgruppe aus.

Nun erfolgt als letztes die Re-Initialisierung der Schleife. Der hier verwendete Befehl 'i++' ist die abkürzende Schreibweise für i=i+1 . Auch das häufig auftretende i=i−1 kann durch i−− abgekürzt werden (x=x+1 −−> x++ usw.).

4. Erhöhe der Wert der Variablen i um eins und gehe zu Schritt 2.

Vergewissern Sie sich bitte an Hand der Schritte eins bis vier, ob Sie den Ablauf innerhalb der 'for'-Schleife verstanden haben. Besonderes Augenmerk verdienen die Punkte eins und drei. Punkt eins wird während der gesammten Abarbeitung der Schleife nur ein einziges Mal am Anfang aufgerufen. Die Besonderheit von Punkt drei ist es, daß die der Schleife zugeordneten Befehle an dieser Stelle ausgeführt werden, sie im Quelltext jedoch erst nach Punkt vier stehen. In Basic ist dies nicht der Fall. Hier steht das annähernd vergleichbare 'next' wie eine Klammer am Ende der der Schleife zugeordneten Befehle (s.o.).

Versuchen Sie nun am Ende dieses Abschnittes über 'for' in C, einige eigene Programme zu schreiben. Verwenden Sie vorerst nur die Vergleichsoperatoren

kleiner	<
größer	>
kleiner gleich	<=
größer gleich	>=

um die oben genannten Fallen zu umgehen. Sehen Sie bitte die folgenden Vorschläge als Anregung:

a. Schreiben Sie ein Programm, das die Zahlen eins bis zehn in umgekehrter Reihenfolge ausgibt.

b. Schreiben Sie ein Programm, das alle Potenzen der Zahl 2 bis zum Wert 1024 ausgibt. (1 2 4 8 16 ... 1024)

Der zweite Teil dieser Kursausgabe wird sich mit der Syntax einer 'while'-Schleife befassen. Im Vergleich zu der eben besprochenen 'for'-Schleife kennt 'while' keine fest vorgesehenen Stellen zum Initialisieren oder Re-Initialisieren von bestimmten Werten.

WHILE

Die Befehle, die von einer 'while'-Schleife zusammengefaßt werden, hängen nur von der zu 'while' gehörenden Bedingung ab. Die Syntax einer Schleifenkonstruktion, die mit 'while' arbeitet, lautet folgendermaßen:

```
while(Bedingung) <Befehl> ;
```

Die durch <Befehl> dargestellte Anweisung an den Rechner wird in Abhängigkeit des für 'Bedingung' eingesetzten Ausdrucks ausgeführt. Ist 'Bedingung' eine WAHRE Aussage, wird der Befehl abgearbeitet, an-

derenfalls wird die Schleife beendet. Auch dieser Befehl ist für Umsteiger von Basic bzw. Pascal nichts neues. Er findet mit

```
10 while Variable = true
    ...
40 wend
```
bzw.

```
while Variable = true do ...
```

seine Entsprechungen. Da alle Befehle innerhalb einer 'while'-Schleife nur von einer Bedingung abhängen, wird diese Art der Schleifenprogrammierung gerne verwendet, wenn das Eintreten der Endbedingung nicht vorher bekannt ist, bzw. bei Schleifenkonstruktionen, die keine logisch zusammenhängende Initialisierung und Re-Initialisierung von Laufvariablen enthalten. Unter Laufvariablen versteht man die Variable, die innerhalb einer Schleife bestimmte Werte durchläuft. (Im oberen 'for'-Beispiel ist dies die Variable 'i'. Gerne werden auch die Buchstaben j, k und n als Variablenname für solche Laufvariablen verwendet.)

Eine typische Anwendung für eine Schleife mit 'while' ist z. B. das buchstabenweise Einlesen eines Namens von der Tastatur. Eine 'for'-Schleife ist in diesem Fall zu unflexibel. Warum? Die Möglichkeiten der Initialisierung einer Variable ist in einem solchen Fall nicht sinnvoll. Auch würde eine nach dem obigen Muster programmierte 'for'-Schleife auf einer genau festgelegten Anzahl von Buchstaben bestehen, d. h. lange Namen werden abgehackt und kurze müssen mit Leerzeichen (Space) aufgefüllt werden. Der Einfachheit halber wollen wir in dem folgenden Beispiel nur Buchstaben einlesen und diese wieder ausgeben. Als erstes folgt die Variante mit 'for':

```
main()
{
/* Variablendefinition */
   int i,c;
/* 10 Buchstaben sollen
    eingelesen werden */
   for(i = 1;i <= 10;i++)
        {
     /* lesen */
           c = getchar();
     /* schreiben */
           putchar(c);
        }
/* Ende */
}
```

Das Beispielprogramm verwendet zwei neue Funktionen. Die Funktion 'getchar()' liefert als Funktionswert das nächste über die Tastatur eingegebene Zeichen, während 'putchar(c)' das Zeichen, das in c gespeichert ist, ausgibt. Diese Funktionen werden später noch ausführlich behandelt. Schauen wir uns jedoch jetzt die Funktionsweise unseres ersten Zeicheneinleseprogramms an. Wir durchlaufen eine Schleife, die von eins bis zehn zählt. Der Grundaufbau ist derselbe wie im 'for'-Beispiel am Anfang, nur wird hier nicht der Wert der Laufvariablen ausgegeben, sondern es wird ein Zeichen eingelesen und dann wieder ausgegeben. Wenn Sie das Programm abgetippt haben und es anstarten, werden Sie gleich den Hauptnachteil der hier verwendeten 'for'-Schleife erkennen. Das Programm erwartet in dieses Form eine feste Anzahl von Zeichen, in unserem Fall zehn. Ein vorheriger Abbruch ist nicht möglich, auch die Eingabe von mehr als zehn Zeichen ist nicht möglich.

Gehen wir nun das Problem mit einer 'while'-Schleife an. Dazu brauchen wir zuerst eine Möglichkeit, das Ende der Eingabe festzustellen. Die Funktion 'getchar()' ist auf einen solchen Fall vorbereitet. Sie liefert einen bestimmten Wert, wenn das Ende der Eingabe erreicht ist. Wir können diesen Wert für unser Programm nutzen.

```
main()
{
   int c;
   c = getchar();
   while(c != EOF)
        {
              putchar(c);
              c = getchar();
        }
}
```

Nach der Ihnen bekannten Variablendefinition folgt die erste Zuweisung der Variablen 'c'. Sie erhält damit den ersten Wert, der von der Bedingung 'c != EOF' überprüft werden kann. Der Operator '!=' testet einen Wert auf Ungleichheit. Die Bedingung kann also wie folgt gelesen werden:

Wenn der Wert von 'c' ungleich der Konstante 'EOF' ist, dann führe die Schleife aus.

Die Konstante 'EOF' ist ein Wert, den die Funktion 'getchar()' zurückgibt, wenn der Benutzer die Tasten <CONTROL> und <Z> gleichzeitig drückt. Dies ist das Zeichen für Ende der Eingabe (EOF = End Of File). Ist die oben genannte Überprüfung überstanden, wird das Zeichen 'c' innerhalb der Schleife mit 'putchar(c)' ausgegeben. Daraufhin wird mit 'getchar()' ein neues Zeichen eingelesen und die Prozedur geht von vorne los.

Haben Sie das Programm eingegeben und übersetzt, werden Sie sehen, daß diese Version um einiges besser funktioniert. Sie können jetzt nach jedem Zeichen die Eingabe abbrechen oder beliebig weitermachen.

Zum Abschluß dieser Folge möchte ich Ihnen noch die Form des Programms vorstellen, die ein fortgeschrittener C-Programmierer verwenden würde:

```
main() { int c;
while((c = getchar()) != EOF) putchar(c); }
```

Ich habe in diesem Beispiel die Abfrage auf EOF und die Eingabe mit getchar() innerhalb der Klammern nach 'while' zusammemgefaßt. Diese Maßnahme spart gegenüber dem oberen Beispiel ein 'getchar()' ein. (Die Variable i, wie sie im 'for' Beispiel verwendet wird, fehlt bei der 'while' Variante sowieso.)

Sie sollten am Ende dieses zweiten Teils den Unterschied zwischen einer 'for'- und einer 'while'-Schleife kennen. Auch sollten Sie nach einiger Übung beide Schleifenarten programmieren können. Die nächste Folge wird sich mit Vergleichsoperatoren sowie dem 'if'- und 'case'-Befehl befassen. Am Ende des dritten Teils folgt dann die erste Zusammenfassung des bis dahin besprochenen Stoffes. Sie haben dann eine Erfolgskontrolle zur Halbzeit. Aber bis dahin wünsche ich erst einmal viel Spaß und Erfolg mit dem zweiten Teil!

(GC)

AMIGA INTERN.



Die Intern-Bücher von DATA BECKER haben für den Computer-Profi schon seit jeher einen ganz besonderen Stellenwert. So kann man beispielsweise C64 Intern selbst in den Reparaturwerkstätten von COMMODORE finden. Ein weiterer, überaus erfolgreicher Internband – ATARI Intern – dient verschiedenen, amerikanischen Softwarehäusern bei der Entwicklung ihrer Software als unverzichtbare Informationsquelle. Fakten, die deutlich machen, was Sie von AMIGA Intern erwarten können. Eben ein Buch vollgepackt mit Informationen, die nur in einem Intern-Buch zu finden sind. AMIGA Intern hilft Ihnen, Ihren Superrechner von Grund auf zu verstehen. Hier werfen Sie quasi einen Blick auf die Innereien Ihres AMIGAs. Unter Anleitung von absoluten Experten versteht sich. Wie sehr Sie hier bis ins kleinste Detail vordringen, verrät bereits ein kurzer Blick ins Inhaltsverzeichnis: Die Hardware des AMIGA mit ausführlicher Beschreibung des 68000-Prozessors, der CIAS, der Costumchips und der Schnittstellen, die Programmierung der Hardware in Assembler mit Speicherbelegung, Interrupts, Grafik und Sound. Doch AMIGA Intern würde diesen Namen nicht verdienen, wenn es zusätzlich nicht noch jede Menge Fakten und Details zum Betriebssystem liefern würde. So erfahren Sie alles Wissenswerte zur EXEC: die wichtigsten Strukturen, Funktion und Arbeitsweise des Multitasking, I/O-Handhabung und Verwaltung der Ressources, EXEC-Base, resetfeste Programme und Strukturen. Fehlt nur noch das AmigaDOS. Auch hierzu nur einige Stichworte: Funktionen, Parameterübergabe, Fehlermeldungen, Bootvorgang, Aufbau der Diskette, IFF-Format, Programmstart von CLI und Workbench, interner Aufbau der CLI-Befehle, die Devices. Selbstverständlich finden Sie auch zahlreiche Beispiele zur Programmierung der EXEC- und DOS-Routinen. Am Ende wissen Sie, warum der AMIGA so enorm leistungsfähig ist – und wir sind gespannt, in welcher Entwicklungsabteilung wir dieses Intern-Buch wiederfinden werden.

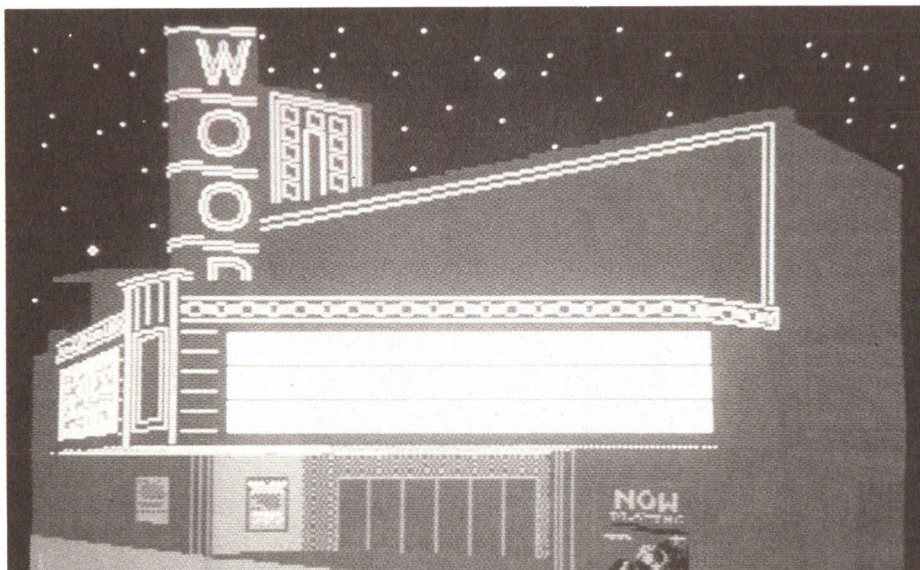
AMIGA Intern
Hardcover, ca. 500 Seiten, DM 69,–
erscheint ca. 9/87

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort

Deluxe Video V 1.2

Ein neuer Video-Generator aus dem Hause Granite Bay Software



Aegis brachte mit dem 3D-Animator eine weitere Fortsetzung eines erfolgreichen Programms. Auch Granite Bay Software will hier nicht zurückstehen. Deluxe Video V 1.2 wurde von den Altmeistern Mike Posehn und Tom Casey geschrieben. Vertrieben wird dieses neue Programm, wie auch Deluxe Video V 1.1, von Electronic Arts.

In diesem Softwarepaket sind vier Disketten enthalten.

Maker Key Disk:

Diese Diskette enthält eine neue Version von Deluxe Video, die sich im Aussehen kaum von der alten Version unterscheidet. Dennoch gibt es Neues an dieser Version, und das nicht zu knapp.

Es ist jetzt möglich, Grafiken, die über den ganzen Bildschirm reichen, zu verwenden. Deluxe Video V 1.2 unterstützt nun auch „Speicherriesen", dies wurde bei bei der Version 1.1 vermisst. Die Möglichkeit, mit einer RAM Disk zu arbeiten, wird jetzt interessant. Die sich daraus ergebenden Geschwindigkeitssteigerungen machen dieses Programm nun im doppelten Sinn zu einem „Renner". Die Verwendung von digitalisierten Soundeffekten wird nun auch mit Deluxe Video V 1.2 zum Vergnügen. Konnte man in der alten Version aus Speichergründen nur kurze Sound Parts verwenden, ist es nun möglich, auch speicherintensive Parts über 28K in seinem Video zu verwenden. Neue Effektmöglichkeiten gibt es auch. Einige neue Foreground-Effekte und einige Löscheffekte machen dieses Programm nun noch mehr „Deluxe".

Player Disk:

Die Player Disk bietet nichts Neues. Sie enthält den Videorecorder, der es ermöglicht, Videos ohne das Hauptprogramm abzuspielen. Durch Anklicken mehrerer Videos kann man auch Videofilme verbinden.

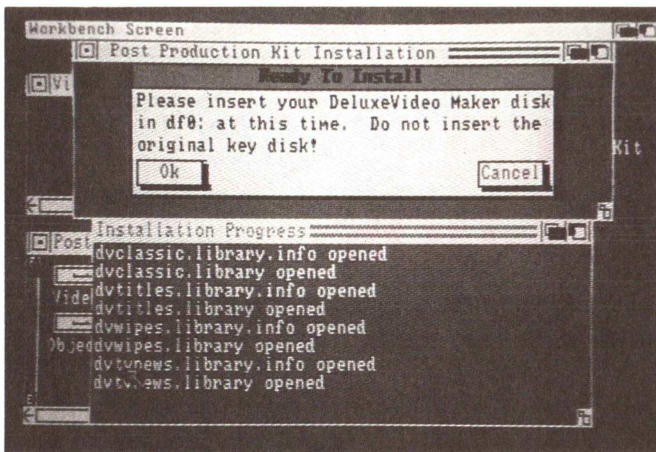
Parts & Utilities Disk:

Diese Diskette ist ebenfalls die gleiche geblieben wie bei Version 1.1. Viele kleinere Programme befinden sich auf dieser Diskette. Mit dem Framer kann man Animationssequenzen aus Grafikprogrammen verwenden (z. B. Cycle Colors). Das Programm Unpack zerstückelt ein Video. Objekts, Bilder und Musikstücke werden aus einem fremden Video herausgenommen und auf Diskette abgespeichert. Nun kann man diese Teile in seinen eigenen Videos verwenden. Vidcheck komprimiert ein Video und fertigt ein Logbuch an. Hier kann man genau lesen, wann und wo welcher Effekt im Video verwendet wurde.

Post Production Kit:

Diese Diskette ist völlig neu. Das Programm auf dieser Diskette nennt sich Install und bedarf einiger Erklärung. Zu dem ausführlichen User Manual liegt ein kleines Heft bei, das dieses Programm und seine Handhabung genau erklärt. Leider, genau wie beim User Manual, ist alles in Englisch.

Die Diskette enthält einige Video Images, die man in seinem eigenen Video verwenden kann. Weiterhin sind auf der Diskette acht neue Bilder, 18 Objekte und drei Sound-Effekte. Mit dem Programm Install wird diese Sammlung von Effekten auf die Maker Disk kopiert und dort





in einem Menü (Scene Generator) verwaltet. Die Effekte werden nun von diesem Selektor aufgerufen und verwendet. Grafiken und Objekte, die nötig sind, um diesen Effekt zu realisieren, müssen vorher geladen werden. Post Production Kit ist auch allein, also ohne Deluxe Video V 1.2, für 30 Dollar erhältlich.

Noch einiges allgemeines zu Deluxe Video V 1.2. und Video Technik auf dem Commodore Amiga: Mit Ausnahme der Maker Disk ist keine der Disketten kopiergeschützt. Da aber gerade mit der Maker Disk sehr viel gearbeitet wird und dadurch die Gefahr eines Disk-Errors sehr hoch ist, ließen sich die Programmierer eine, wie ich meine, sehr gute Kompromißlösung einfallen. Beim Laden des Programmes Deluxe Video von einer Kopie wird nach einem kurzen Ladevorgang die Orginal Disk verlangt. Nun wird das Programm gebootet und man kann wieder die Kopie einlegen.

Für Anwender mit nur einem Laufwerk kann das Arbeiten mit Deluxe Video zu Qual werden. Gerade beim Laden von Objekten und Bildern ist das Wechseln der Disketten nötig. Auch das Arbeiten mit der RAM-Disk ist hier keine befriedigende Lösung, denn nicht jeder hat 1 MByte zur Verfügung.

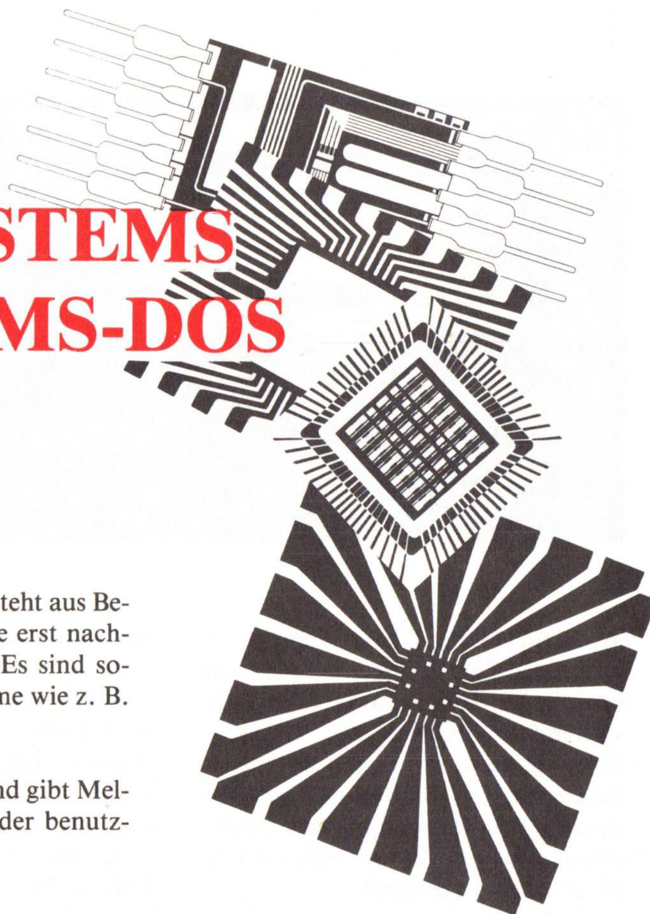
Mit Programmen wie Aegis Animator, 3D-Animator, Amiga Art Machine, Deluxe Video V 1.1 und V 1.2 wird das Arbeiten mit dem Amiga auf dem Gebiet der Videotechnik noch professioneller. Die Programme nutzen die Hardware des Amiga aus, und diese Fähigkeiten erheben den Amiga schon deutlich über den Standard eines Homecomputers. Die Hardwareerweiterungen machen dieses System noch leistungsstärker. Hier Digitalisiergeräte für Bilder und Sound und nicht zuletzt Genlock-Interfaces zu nennen. Die einfache Handhabung des Amiga und der Software macht es auch für Laien leicht, profesionelle Videoclips herzustellen. Die Einsatzmöglichkeiten dieser neuen Videotechnik sind unbegrenzt. Von kleinen Videos, nur auf dem Computer, über das Bearbeiten von Hobbyfilmer-Produktionen bis hin zu professionellen Music-Clips ist alles möglich. Man kann nur abwarten, was die Hard- und Software-Tüftler noch erfinden, um aus jedem einen kleinen Steven Spielberg zu machen!

Thomas Schlereth

Bezugsquelle:
DTM, Wiesbaden
Tel: 0 61 21 / 56 00 84

Preis: DM 249,–

ENTWICKLUNG EINES BETRIEBSSYSTEMS AM BEISPIEL VON MS-DOS



Anfang der achtziger Jahre entschied sich IBM, einen Personal-Computer auf den Markt zu bringen. Das Betriebssystem sollte PC-DOS (MS-DOS, MicroSoft Disk Operating System) sein.

MS-DOS ist ein sehr preiswertes Betriebssystem, das man für wenige hundert Mark erwerben kann. Es besteht aus den Teilen

1. IBMBIO
2. IBMDOS
3. COMMAND.COM

Die Dateien IBMBIO und IBMDOS sind 'hidden files', d. h. sie werden nicht im Inhaltsverzeichnis angezeigt. IBMBIO ist der physikalische Gerätetreiber und IBMDOS ist der logische Treiber. COMMAND.COM wird im Directory der Diskette angezeigt und ist der Console Command Processor.

MS-DOS ist in zwei Teile aufgeteilt, einen transistenten und einen residenten. Der residente Teil befindet sich im COMMAND.COM. Zu diesem Teil gehören Befehle wie

DIR
druckt das Inhaltsverzeichnis der Diskette an,

DATE
zeigt oder ändert das Datum,

TIME
zeigt oder ändert die Uhrzeit,

DEL
löscht die anzugebende Datei,

COPY
kopiert eine Datei auf eine andere.

Der transistente Teil besteht aus Betriebssystem-Befehlen, die erst nachgeladen werden müssen. Es sind sogenannte Dienstprogramme wie z. B.

CHKDSK
überprüft eine Diskette und gibt Meldungen über die Anzahl der benutzten Bytes an,

DISKCOMP
vergleicht zwei Disketten, ob sie dieselben Daten gespeichert haben,

FORMAT
formatiert eine Diskette neu,

DISKCOPY
kopiert eine Diskette auf eine andere,

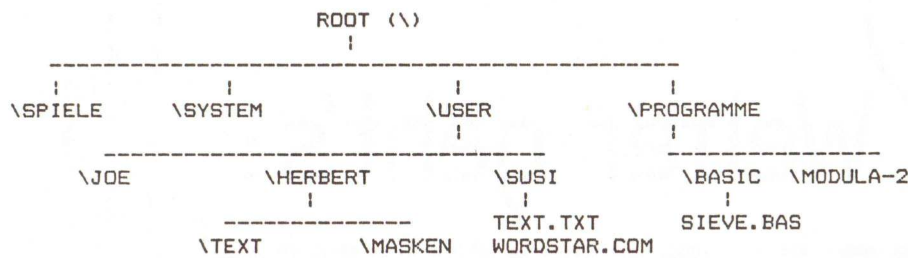
EDLIN
Zeilen-Editor, um ein ASCII-File zu erstellen,

LINK
erzeugt eine .EXE (executable) File von einem .OBJ (object code) File.

MS-DOS ist in seinen ersten Versionen sehr an das damals weit verbreitete Betriebssystem CP/M (Control Program for Microcomputer) angelehnt. Bei der MS-DOS-Version 1 existieren Befehle wie RDCPM, um eine Diskette zu lesen, die mit CP/M formatiert wurde. Zusätzlich wurden Befehle aus Kompatibilitätsgründen doppelt belegt wie REN und RENAME (Datei umbenennen) oder DEL und ERASE. Die zweitgenannten Kommandos sind in CP/M vorhanden.

Die große Ähnlichkeit zwischen diesen beiden Betriebssystemen hatte für MS-DOS den großen Vorteil, daß es sehr schnell populär werden konnte, da viele Programmme, die für CP/M geschrieben wurden, mit geringem Aufwand für die Software-Entwickler an MS-DOS angepaßt werden konnten. Die bekanntesten Beispiele hierfür sind wohl WORDSTAR (ein Textverarbeitungssystem) und Turbo Pascal (eine Programmiersprache). Die zu MS-DOS mitgelieferte Programmiersprache BASIC ist auch fast identisch zu dem damals für CP/M erhältlichen MBASIC von MicroSoft. MS-DOS ist ein Betriebssystem für 16-Bit-Microcomputer, CP/M dagegen eines für 8-Bit-Computer. Als nennenswerte Alternative läßt sich für 16-Bitter lediglich noch CP/M 86 erwähnen, eine konstante Weiterentwicklung von CP/M 80 bzw. CP/M 85.

Das war der Stand Anfang der achtziger Jahre. Doch die Entwicklung ging weiter. Als nächstes kam von MicroSoft die Version 2.xx von MS-DOS auf den Markt. Die Erweiterungen gingen von CP/M weg in Richtung UNIX und dessen Derivate XENIX und SINIX. Das fängt zum Beispiel an bei der Möglichkeit, das Directory hierachisch zu gliedern. Ein Inhaltsverzeichnis einer Diskette sieht nun z. B. so aus:



Mit ROOT bezeichnet man das Ursprungsdirectory. Von ihm aus starten alle Verzweigungen in Subdirectories. Es gibt ab Version 2.0 folgende Befehle zur Verwaltung dieser Directory-Struktur:

MkDir Make Directory
erzeugt ein neues Subdirectory.

RmDir Remove Directory
entfernt ein Subdirectory (nur leere Subdirectories können entfernt werden).

ChDir Change Directory
Wechselt von einem Directory zu einem anderen.

Um zum Beisiel das Programm WORDSTAR aufzurufen, gibt es zwei Möglichkeiten. Einmal kann man den genauen Pfad angeben, zum anderen könnte man in das Subdirectory USER\SUSI wecheln. Die Befehle würden dann so aussehen:

A: USER\SUSI\WORDSTAR oder
A:CHDIR USER\SUSI
A:WORDSTAR

Wie bei UNIX trennt der Backslash '\' die einzelnen Directories voneinander. Diese umständliche Prozedur kann man sich mit Hilfe des PATH (Pfad)-Befehls erleichtern. Für häufig verwendete Programme, die in Subdirectories stehen, kann man MS-DOS einen Suchpfad vorgeben, in dem dann das Programm zusätzlich zu der ROOT noch gesucht wird. Beispiel:

A:PATH= USER\SUSI
A:WORDSTAR

Der angegebene Suchpfad gilt solange, bis er geändert oder das Gerät ausgeschaltet wird. Mehrer Suchpfade sind möglich, sie werden durch ein Semikolon getrennt. Beispiel:

A:PATH USER SUSI; USER HERBERT MASKEN; SYSTEM

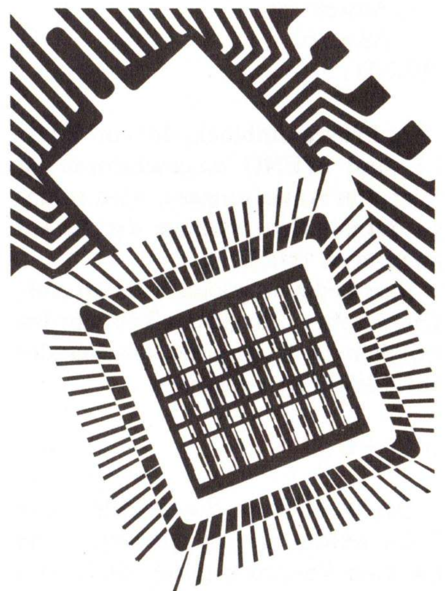
Mit Hilfe dieser Befehle kann man Disketten oder Festplatte sehr gut in Ordnung halten. Die nächsten Versionen von MS-DOS haben die Nummer 3.1 und 3.2. Ab diesen Versionsnummern ist es möglich, MS-DOS im Netzwerk zu fahren. IBM bietet ein TOKEN-RING-Netzwerk an, um PCs miteinander zu verbinden. Die Vorteile eines Netzwerkes liegen auf der Hand, z. B. die Vervielfachung der Plattenkapazität, denn nicht jeder angeschlossene PC benötigt eine Festplatte. Daten müssen nicht redundant (mehrfach) abgespeichert werden und anderes mehr. Neben dem TOKEN-RING-Netzwerk gibt es noch andere topologische Möglichkeiten, Netzwerke aufzubauen. Zu erwähnen währe hier noch das Stern-Netzwerk, das einem Großrechner sehr ähnelt, und das Busnetzwerk. MS-DOS bietet hier jedoch lediglich den SHARE-Befehl an, um eine Datei für mehre Benutzer zugänglich zu machen. Den Rest muß dann eine Netzwerk-Hard- und Software erledigen. Als Erweiterung gegenüber Version 2.xx sind noch die Befehle JOIN und SUBST zu nennen. Der JOIN-Befehl erlaubt die absolute hierachische Verwaltung von Dateien. Mit ihm ist es möglich, Laufwerksnamen durch Subdirectorynamen zu ersetzen. Durch das Kommando A:JOIN B: A: DISKB wird das logische Laufwerk B: zu dem Subdirectory DISK-B. Der SUBST-Befehl bewirkt genau das Umgekehrte. Mit ihm kann man ein Subdirectory zu einem logischen Laufwerk machen. Der Befehl SUB-ST E: B: USER SUSI bewirkt, daß aus dem Subdirectory USER SUSI das logisches Laufwerk E: wird. Dieser Befehl ist sehr nützlich, wenn man noch alte CP/M-Programme benutzt, die nicht das hierachische Dateiensystem unterstützen. Beispiele hierfür sind wieder WORDSTAR und Turbo Pascal Ver.2.0 und früher. Eine Neuerung, die jedoch erst von MS-DOS 3.2 unterstützt wird, ist der DSKSETUP Befehl, der es erlaubt, ein Diskettenlaufwerk zu konfigurieren. Grundsätzlich muß man nur angeben, welches der drei möglichen Laufwerke man besitzt. 5,25 Zoll mit 360 KB, 5,25 Zoll mit 1.2 MB oder 3,5 Zoll mit 720 KB.

Dies soll als kurzer Überblick über die Entwicklung von MS-DOS genügen. Doch wohin geht es weiter? IBM hat vor kurzem eine neue Rechnerserie herausgebacht. Sie nennt sich Pesonal System /2. Auf diesen Rechnern gibt es zur Zeit nur das Betriebsystem PC-DOS 3.3, wieder eine Weiterentwicklung des guten alten MS-DOS. Als nächstes ist System OS/2 (Operating System No.2) angekündigt. Es erlaubt dann den Multi-Tasking-Betrieb, d. h. es können mehrere Programme quasi gleichzeitig ablaufen. (Der AMIGA erlaubt den Multi-Tasking-Betrieb mit Hilfe von AMIGADOS schon jetzt).

Die Frage, ob OS/2 auf dem AMIGA mit SideCar oder mit XT- bzw. AT-Board läuft, ist in Moment noch offen. MicroSoft teilte jedoch mit, daß OS/2 auf jedem AT lauffähig sein soll.

Roland Foerster

Bibliographie
MS-DOS REFERENCE GUIDE
VERSION 1.xx, 2.xx, 3,xx





Weiter geht's: AMIGA-BASIC-KURS Teil 2

In der Hoffnung, daß Sie sich inzwischen mit dem Stoff des ersten Teils beschäftigt haben, wollen wir hier fortfahren. Es sind noch einige grundlegende Basic-Befehle und Eigenschaften zu erklären, bis auch die Anfänger einen Überblick haben. Erst dann kann man sich mit den besonderen AMIGA-Leckerbissen wie Intuition-Programmierung mit Hilfe der Libraries beschäftigen.

Der letzte Teil endete bei den Schleifen. Da AMIGA-Basic glücklicherweise noch eine weitere Schleife besitzt, werden wir diesen Teil damit beginnen. Gemeint ist die WHILE.. WEND-Schleife. Sie sieht so aus:

```
WHILE Bedingung
      Anweisung
      Anweisung
      Anweisung
WEND
```

Der Programmblock, der von WHILE und WEND eingeschlossen ist (hier drei Anweisungen), wird solange durchlaufen, solange die Bedingung noch erfüllt ist.

Eine Bedingung kann vielerlei Aussehen haben. Im ersten Teil wurden lediglich zwei Variablen miteinander auf Gleichheit verglichen.

```
IF A=B THEN ....
```

Der Befehl bzw. die Befehle hinter Then werden nur ausgeführt, wenn die zwei Variablen gleich sind. Nun gibt es natürlich Situationen, bei denen dies nicht ausreicht, beispielsweise die Abfrage, ob eine Zahl größer als die andere ist. Dazu bietet Basic mehrere Möglichkeiten. Also alles der Reihe nach.

```
IF A<B THEN ..
```

Hier wird geprüft, ob A kleiner ist als B.

Bezogen auf die WHILE-Schleife sähe dies so aus:

```
WHILE Leitung_besetzt
      Ärgern
      Neu_Wählen
WEND
Telefoniere
Lege_Hörer_auf
```

Der Sinn ist klar: Solange die Leitung besetzt ist, muß man erneut wählen. Ist sie dann frei, telefoniert man und legt anschließend wieder auf.

Solche Abfragen kommen natürlich nicht nur bei Schleifen vor, sondern besonders bei IF..THEN..ELSE.

Nehmen wir folgendes kleines Beispiel:

```
LET Kinopreis = 8.50
INPUT "Wieviel Geld haben Sie noch";Geld
IF Geld < Kinopreis THEN
   PRINT "Zu wenig Geld, mein Freund"
   PRINT "Wie wäre es mit Fernsehen ?"
ELSE
   PRINT "Ihre Karte, werter Gast"
   PRINT "Darf ich Ihnen Ihre Loge zeigen"
END IF
```

Anhand dieses Beispiels sieht man, daß man bestimmte Probleme recht eindeutig (fast schon in Klartext) programmieren und sehr leicht anpassen kann. Nehmen wir an, jemand kann Kino nur mit einer Pakkung Eiskonfekt ertragen. Dann muß er noch zwei Mark mehr aufbringen. In unserem Programm ließe sich das durch Änderung einer Zeile bewerkstelligen.

```
      ..
IF Geld < (Kinopreis+2) THEN
      ..
```

Der Basic Interpreter addiert die Zwei zum Kinopreis und überprüft erst dann, ob man sich den Kinopreis eigentlch leisten kann.

Die möglichen Abfragen im Zusammenhang:

=	Gleich
<	Kleiner
>	Größer
<>	Ungleich
<=	Kleiner oder gleich
>=	Größer oder gleich

Sie sehen, daß man damit schon eine Menge anfangen kann; aber probieren Sie es doch einfach einmal aus. Manchmal kommt es jedoch vor, daß man mehrere Abfragen kombinieren muß, z. B. folgende:

Wenn es gilt, mehrere Bedingungen abzufragen, kann man die einzelnen Abfragen auch kombinieren. Ein

Beispiel aus dem täglichen Leben verdeutlicht dies.

```
WENN Arbeit_zuende UND
Schönes_Wetter DANN
Gehe_Ins_Schwimmbad
```

Man geht also nur ins Schwimmbad, wenn schönes Wetter ist und auch Feierabend. Ist eine der beiden Bedingungen nicht erfüllt, so kann man die Badehose eingepackt lassen. In Basic geht dies mit der AND-Verknüpfung vonstatten. Umgesetzt könnte es folgendermaßend aussehen:

```
IF Zeit> =16 AND Temp>25
THEN PRINT "Ich geh ins
Schwimmbad"
```

Ein weiteres Beispiel:

```
WENN Kein_Benzin ODER
Platter_Reifen DANN
Fahre_mit_Straßenbahn
```

In Basic hieße es wie folgt:

```
IF Menge_Benzin=0 OR
Reifendruck<1 THEN PRINT
"Hallo S-Bahn"
```

Man sieht, daß die Ähnlichkeit zur Sprache vorhanden ist.

Auch komplexere Probleme lassen sich damit lösen. Ein Beispiel:

```
WENN Montag UND
Auto_nicht_kaputt UND
Noch_Benzin ODER
Lust_zu_laufen UND
Kein_Regen DANN
Gehe_ins_Kino ANSONSTEN
Bleibe_ich_zuhause
```

Diese Verknüpfung ist schon etwas schwieriger, denn sie hat – obwohl der Sinn klar ist – ihre Tücken. Es gibt zwei Möglichkeiten, ins Kino zu kommen: Mit dem Auto, aber nur, wenn noch Benzin im Tank ist, oder zu Fuß, das aber nur bei gutem Wetter. Das schönste kommt noch: Man will natürlich nur Montags ins Kino, weil da der Eintritt billiger ist. Das mag zwar etwas wirr klingen, aber will man dies in Basic übersetzen, dann muß man schon die Klammerung beachten. Also frisch ans Werk:

```
IF Montag AND ( (Auto< >Kaputt
UND Benzin<0) OR (Lust_zu_
laufen AND Regen=0) ) THEN
   Print "Kino ich komme"
ELSE
   Print "Ich bleibe lieber zuhause"
   Print "Schade "
END IF
```

Man kann dies natürlich noch ins Unendliche weitertreiben (was ist, wenn man den Film schon kennt ?), aber jetzt ist Schluß. Auch wenn diese Beispiele etwas aus dem Leben gegriffen sind, zeigen sie doch recht deutlich die Möglichkeiten, ein Problem umzusetzen.

Funktionen

Unter den Begriff Funktionen fallen in Basic viele Befehle, genauer gesagt 47 (laut Handbuch). Doch was sind Funktionen eigentlich? Dazu einige Beispiele:

```
X=SQR(2)
PRINT SIN(3.1415)
Zeit=TIMER
A$="Holarödulíöh" :
PRINT LEN(A$)
```

In der ersten Zeile wird die QUADRATWURZEL von der Zahl 2 gezogen und der Variablen X zugeordnet. Quadratwurzeln braucht man recht oft, beispielsweise bei der Berechnung der Diagonalen eines Quadrates. In der zweiten Zeile wird der SINUS einer bestimmten Zahl berechnet und sofort auf dem Bildschirm ausgegeben. Man sieht, daß der errechnete Wert nicht erst einer Variablen übergeben werden muß, sondern auch direkt verarbeitet werden kann. Der Befehl in der dritten Zeile sorgt dafür, daß die Systemzeit des Rechners der Variablen Zeit zugeordnet wird. TIMER ist auch nichts anderes als eine Funktion, eine recht brauchbare sogar, denn mit ihr hat man stets die Zeit zur Hand. Im letzten Beispiel wird schließlich die LÄNGE eines Strings ausgedruckt. Dazu wurde der String zuerst einmal definiert, da sonst die Länge unweigerlich Null wäre. So erhalten wir das Ergebnis '12', da besagter Freudesschrei aus zwölf Buchstaben besteht.

Die Möglichkeiten, die mit Funktionen angeboten werden, sind relativ groß. Auch hier ist Probieren angesagt, ein Blick ins Handbuch verrät mehr.

Aber damit nicht genug; es besteht weiterhin die Möglichkeit, eigene Funktionen zu kreieren. Nehmen wir ein Beispiel aus der Mathematik. Man braucht zum Beispiel, aus welchen Gründen auch immer, den Arcussinus einer Zahl. Tja, normalerweise schlechte Karten, da AMIGA-Basic diese Funktion nicht kennt. Dies ist aber noch lange kein Grund, gleich aufzugeben und einen anderen Rechner zu kaufen. Man definiert ihn einfach selbst. Dazu nimmt man den Befehl DEFFN zur Hilfe.

```
DEFFN ARCSIN(X)=ATN(X/SQR
(1-X*X))
```

Einfach sieht das wohl nicht gerade aus, aber ganz so schlimm ist es auch wieder nicht. Der Name hinter DEFFN, in diesem Fall sinngemäß ARCSIN, ist der Name der neuen Funktion. Die Variable in Klammern zeigt, daß diese Funktion einen Parameter übergeben bekommt.

Der Befehlwirrwar hinter dem Gleichheitszeichen berechnet den Arcussinus der Zahl X. Wie er zustande kommt, soll uns hier nicht weiter stören.

```
PRINT FN ARCSIN(.57)
```

Um noch der Mathematik ein paar Zeilen zu schenken, überprüfen wir noch schnell, ob die Funktion auch richtig rechnet. Der Arcussinus ist bekanntlich die Umkehrfunktion des SINUS. Also berechnen wir von einer Zahl den Sinus und von diesem Ergebnis den Arcussinus. Kommt dabei die Zahl wieder heraus, so war die neue Funktion richtig. Also:

```
PRINT FN ARCSIN( SIN(1.2345))
```

Das Ergebnis lautet 1.2345 – der Beweis.

Die eben definierte Funktion bekam einen Parameter übergeben, man kann eine Funktion aber auch so gestalten, daß sie keinen oder gleich mehrere Parameter bekommt. Zwei

```
DEF FN Arcsin(x)=ATN(x/SQR(1-x*x))
PRINT  FN Arcsin(SIN(1.2345))

DEF FN PI=4*ATN(1)
PRINT FN PI

DEF FN Volumen(hoch,radius) =radius*radius * FN PI * hoch
PRINT  FN Volumen(3,2)

DEF FN Wuerfel   = INT(RND(1)*6)+1
RANDOMIZE (TIMER)
FOR i=1 TO 10
  PRINT  FN Wuerfel;
NEXT i
```

Parameter benötigt die im Listing definierte Funktion VOLUMEN. Sie berechnet das Volumen eines Zylinders.

Die Funktion WUERFEL hingegen benötigt keine Werte, sondern liefert, nach dem Motto besser Geben als Nehmen, eine Zufallszahl zwischen Eins und Sechs. Das Ergebnis läßt sich leicht prüfen, indem man sich einfach einige Würfelzahlen ausdrucken läßt. Der etwas seltsam anmutende Befehle 'Randomize Timer' sorgt dafür, daß bei jedem Start der Zufallsgenerator andere Werte liefert. Das ist wichtig, da ansonsten immer gleiche Zahlen kommen (man will schließlich als ehrlicher Mensch nicht mogeln).

Unterprogramme

Wir kommen jetzt zu einer sehr wichtigen Fähigkeit des AMIGA-Basic: Zu den Unterprogrammen. Doch was soll man mit Unterprogrammen eigentlich anfangen, genügt nicht ein Hauptprogramm?

Unterprogramme sind dazu da, Programme zu strukturieren, also übersichtlich zu gestalten. Sie können dadurch die Arbeit sehr vereinfachen.

Ein Unterprogramm wird in AMIGA-Basic mit SUB gekennzeichnet, genauer gesagt mit SUB Name. Name ist der Name der SUBroutine, mit dem man das Unterprogramm dann aufruft. Die nachfolgenden Parameter werden dem Unterprogramm übergeben. Dort werden sie verarbeitet. Dies kann sich über mehrere Befehlszeilen hinstrecken. Falls erwünscht, bekommt man die errechneten Werte wieder zurück.

Beginnen wir mit einem einfachen Beispiel:

```
SUB Hallo Static
   PRINT "Hallo Meister"
   PRINT "Guten Morgen"
 END SUB
```

Im Programm nun wird das Unterprogramm einfach mit dem Wort 'Hallo' aufgerufen. Auf dem Bildschirm erscheint sofort der obige Text. So ein Unterprogramm sollte man immer dann anwenden, wenn bestimmte Anweisungen öfter gebraucht werden, man spart sich eine Menge Arbeit.

Man kann einem solchen Unterprogramm auch Werte übergeben, sozusagen zur Bearbeitung, und erhält den Wert bearbeitet zurück. Folgendes Beispiel dreht einen beliebigen String um. Dies wird mit einer Schleife bewerkstelligt, die von Eins bis zur Länge des Strings läuft, und zwar rückwärts. In unserem Beispiel von 14 bis 1. Der Befehl MID$ sorgt dafür, daß der Reihe nach (von hinten nach vorne) jeweils ein Buchstabe herausgepickt wird und zur Variablen Rueck$ dazukommen. Ist die Schleife beendet, kehrt Basic, veranlasst durch den Befehl 'END SUB', ins Hauptprogramm zurück. Der Aufruf geschieht durch den Befehl

```
Mirror Jodel$,Erg$
```

Jodel$ enthält den frohen Grußesruf – In Erg$ bekommt man den gedrehten Ruf geliefert.

Neben Stringvariablen kann man den Unterprogrammen natürlich auch Zahlenvariablen übergeben bis hin zu kompletten Feldern.

Hat man erst einmal an Unterprogrammen Gefallen gefunden, wird man auf sie sicher nicht mehr verzichten wollen, schließlich erlauben sie das Gestalten von Basic-Befehlen, die ganz an den eigenen Bedarf angepasst werden können. Wenn das nichts ist!

Zum Ende dieses Teil noch einmal der Appell, sich unbedingt mit den erwähnten Befehlen zu beschäftigen, viel auszuprobieren und auch einen Blick in ein Basic-Lehrbuch zu werfen, da wir hier leider die Möglichkeiten dieser Sprache nur kurz anreißen können. Und noch ein Tip: Programmieren zu zweit geht grundsätzlich besser!

HS

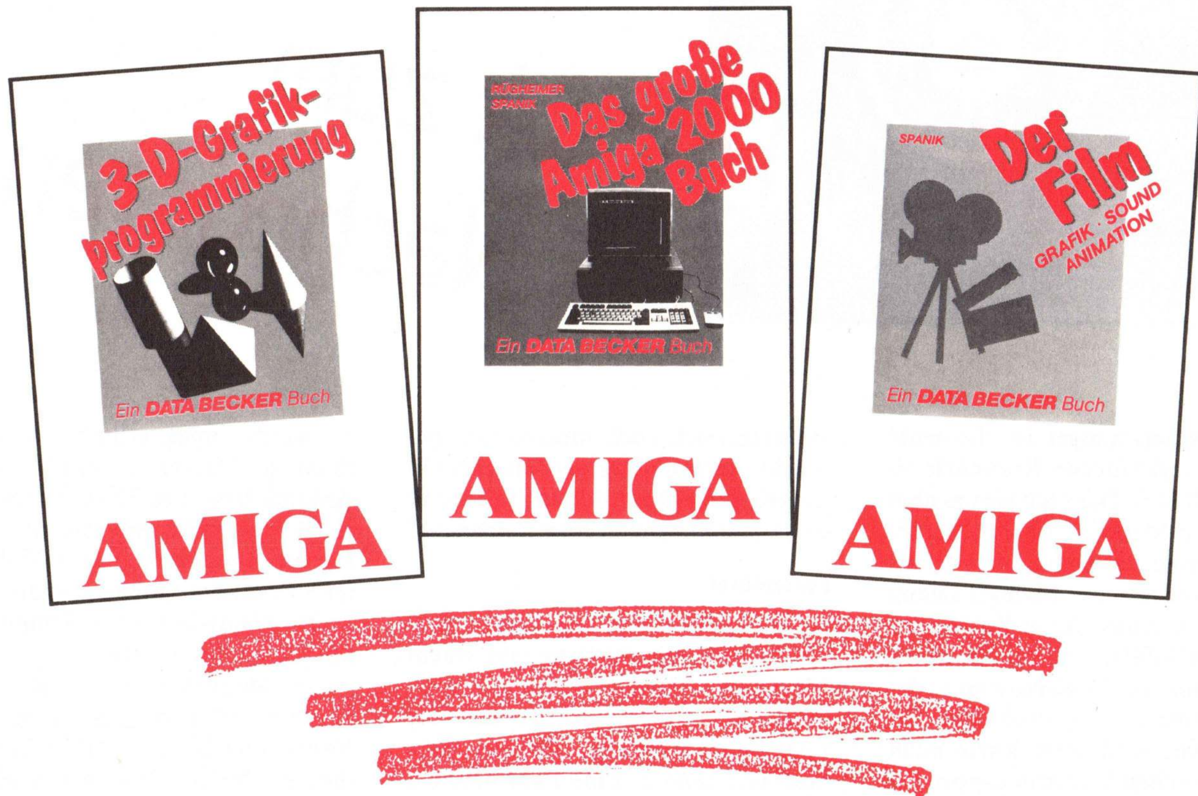
```
jodel$="Holdrijo-dulie"
PRINT jodel$
Mirror jodel$,erg$
PRINT erg$

STOP

SUB Mirror (vor$,rueck$) STATIC
' Dreht dem Jodler das Wort im Munde um
  FOR i=LEN(vor$) TO 1  STEP-1
    rueck$=rueck$+MID$(vor$,i,1)
  NEXT i

END SUB
```

Drei auf einen Streich:



3-D-Grafik auf dem Amiga - hier ist der Schlüssel zu dieser faszinierenden Welt. In diesem Buch werden Grafikalgorithmen beschrieben und erläutert, die es in sich haben. Mit ihnen können Sie absolut realistisch gestaltete Bilder erstellen: Die einzelnen Komponenten eines Bildes lassen sich dabei mit einem Editor problemlos eingeben und solange durch die Wahl verschiedener Materialien, Farben, Lichtquellen und Spiegelungen verfeinern, bis sie eine absolut naturgetreue Darstellung erreicht haben. Die Berechnung des Bildes übernimmt das Programm automatisch, in allen Auflösungen mit bis zu 4096 Farben. 3-D-Grafikprogrammierung - ein Muß für alle, die Spaß an tollen Grafiken haben.

Amiga 3-D-Grafikprogrammierung
Hardcover, ca. 250 Seiten
inkl. Diskette, DM 59,-
erscheint ca. 9/87

Läßt das Handbuch Sie auch in so manchen Dingen allein, das große Amiga-2000-Buch nicht. Dieses Buch wird schnell Ihr ständiger Begleiter bei der Arbeit mit dem Amiga 2000 sein. Hier finden Sie eine umfassende Einführung in die Arbeit mit Ihrem neuen Rechner - und mehr als das. Denn sind Sie erst einmal mit dem Amiga 2000 „per Du", zeigen Ihnen die Autoren, was einen Amiga-Profi ausmacht: Laufwerkeinbau in den Amiga 2000, Speichererweiterung, Arbeiten mit der PC-Karte, Datenaustausch zwischen Amiga und PC, Einbau und Einrichtung einer Harddisk, Karten für Amiga- oder PC-Seite, PC-Audioausgänge, erste Hilfe bei Harddisk-Abstürzen, Kickstart im RAM, die richtige Software für Ihren Erfolg, und und und. Sollten Sie also noch Fragen zu Ihrem Rechner haben, das große Amiga-2000-Buch liefert Ihnen die Antworten.

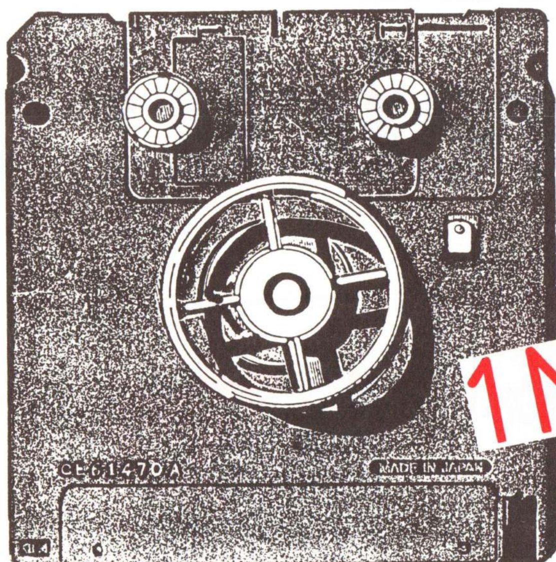
Das große Amiga 2000 Buch
Hardcover, 684 Seiten, DM 59,-

Der Amiga ist das ideale Werkzeug für Kreative - aber einen Film mit ihm erstellen? Warum nicht? Dieses Buch hilft Ihnen dabei. Vom Drehbuch bis zum fertigen Drei-Minuten-Film. Mit allem, was dazu gehört: Grafik, Sound, Animation und und und. Dabei arbeiten Sie mit den verschiedenen DeLuxe-Programmen - auch mit DeLuxe-Video 1.2 und DeLuxe-Paint II, lernen deren fantastische Möglichkeiten und Anwendungsgebiete kennen. Da Sie als angehender Filmemacher auch mit Videorekorder und Sounddigitizer professionell umgehen müssen, wird das nötige Know-how gleich mitgeliefert. Wann dürfen wir zu Ihrem ersten Oscar gratulieren?

Amiga - Der Film
Hardcover, ca. 400 Seiten
DM 49,-
erscheint ca.
Ende 10/87

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,- Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort



Speicher satt!

1 MEGA zum Nachrüsten

Speicherplatzmangel ist die wohl am meist verbreitetste Krankheit einiger AMIGAS. Dagegen gibt es aber bekanntlich Medizin. Wir wollen Ihnen hier eine davon vorstellen: die MegAmiga Karte von Michael Lamm Computersysteme. Diese Karte stattet Ihren A 1000 mit einem Megabyte Speicher aus. Im Gegensatz zu anderen für diesen Rechner erhältliche Erweiterungen, wird diese Karte nicht in den seitlichen Expansionsport des Amiga gesteckt, sondern findet an Stelle der 256 Kbyte Karte unter der Frontabdeckung ihren Platz.

Die Erweiterung ersetzt bei ihrem Einbau eine eventuell schon vorhandene 256 Kb Karte (bei den neuen A 1000 ist diese Karte bereits im Lieferumfang enthalten). Sie enthält 24 Rambausteine des Typs 41464. Acht dieser ICs ersetzen die 256 Kbyte der original Commodore Karte. Die zusätzlichen 16 Ram ICs ergeben den von der Karte versprochenen 1 Megabyte großen Speicher.

Der Entwickler der Karte hat dem Endbenutzer die Möglichkeit gegeben den Adressraum des zusätzlichen Speichers wahlweise auf zwei Bereiche einzustellen. Als erster Bereich sind die Adressen $080000 – $0fffff möglich. Ihr Amiga verfügt bei dieser Konfiguration über ein Megabyte linear adressierbaren Speicher (keine Lücken im Adressraum). Die zweite Beschaltung legt die zusätzlichen 512 Kbyte auf $180000 – $1fffff. Durch das Anbringen eines zusätzlichen, nicht im Lieferumfang enthaltenen, Schalters (1 mal Um) kann der Adressbereich auch umschaltbar gemacht werden. Das als 'Handbuch' mitgelieferte Din A4 Blatt beschreibt diese Option anhand einer Zeichnung.

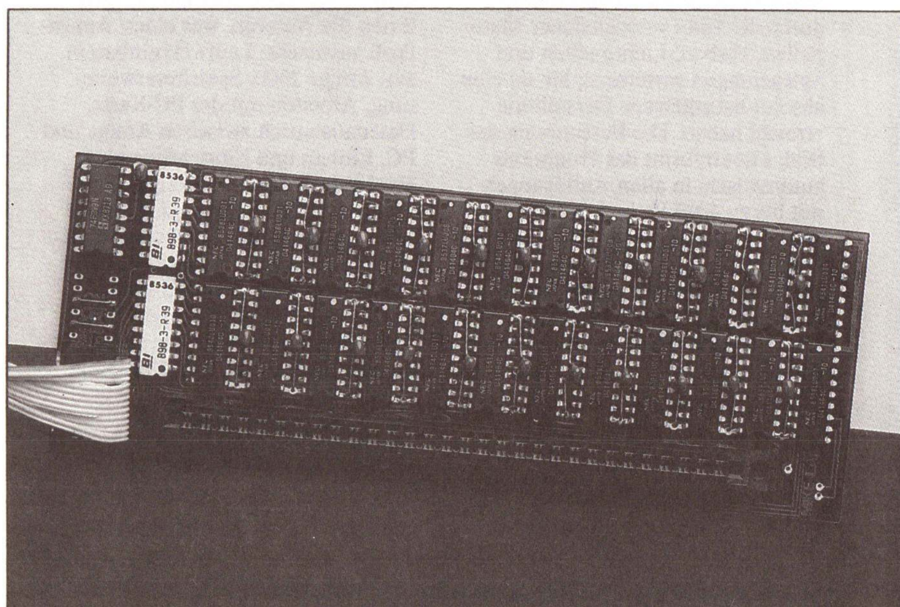
Festgelötet

Da am Frontport des Amiga 1000 die für die Adressierung von einem Megabyte nötigen Adressleitungen nicht vorhanden sind, enthält die Ram Karte einen 14 poligen DIL-Stecker (Dual in Line) auf dem ein ca. 15 cm langes Kabel aufgesteckt ist. Mit diesem Kabel werden die benötigten Signale aus dem Rechner abgegriffen. Der Käufer dieser Karte sollte dazu über ein wenig Lötgeschick verfügen, da insgesammt 14 Leitungen an vier ICs der Hauptplatine gelötet werden müssen. Da zu diesem Vorgang der Rechner geöffnet werden muß, erlischt bei Neugeräten die Garantie. Auch ist das Anlöten bzw. das Herausbiegen von IC Beinen nicht jedermanns Sache. Für einen mittelprächtigen Hobbylöter bereitet der Einbau allerdings keine Probleme und ist in knapp einer Stunde zu bewältigen.

Die MegAmiga läuft mit beiden zur Zeit existierenden Amiga 1000 Versionen (die neue Pal Version und die alte NTSC Version mit Huckepackplatine). Auf dem mitgelieferten Handzettel ist der jeweils verschiedenartige Lageplan, der oben erwähnten vier ICs enthalten, so das lästiges Suchen nach den richtigen ICs entfällt.

Kickstart V 1.1 und

Die Karte wird von der System-



software V 1.1 beim Booten nicht erkannt und muß daher mit Hilfe des ADDMEM Befehls in das System eingebunden werden. Als Parameter benötigt der ADDMEM Befehl die untere und obere Adresse des zusätzlich einzubindenden Speichers. Diese Werte sind entsprechend der oben erwähnten zwei Adressraummöglichkeiten einzusetzen.

Kickstart V 1.2

Die Kickstartsoftware V1.2 erkennt zwar den zusätzlichen Speicher, hält ihn aber fälschlicherweise für 'CHIP-MEM'. Dieser Umstand kann durch ein MAKE512 in der Startup-Sequence rückgängig gemacht werden. Die Einbindung muß danach ebenfalls mit ADDMEM erfolgen.

oder MegaKick V 1.2

Der Hersteller der Karte liefert jedoch auch eine geänderte Kickstart Version auf Diskette die die Karte automatisch erkennt und richtig in das System einbindet.

Fazit

Der nicht gerade Lötneuling erhält mit dieser Karte eine vergleichsweise billige und flexible Speichererweiterung. In der bis jetzt kurzen Testphase der Karte ergaben sich keine Probleme mit der Software, zumal die MegAmiga Karte auch abgeschaltet werden kann (ebenfalls über einen Schalter 1 mal Um).

(GC)

+ keine Platzprobleme
+ frei konfigurierbar

– es muß gelötet werden (Garantie erlischt)
– Treibersoftware nicht im Lieferumfang enthalten

Hersteller und Vertrieb:
Michael Lamm Computersysteme
Schönbornring 14
6078 Neu-Isenburg 2
Tel.: 06102/52535

Preis:
MegAmiga Karte ca. 398,– DM
MegaKick Generator Disk V 1.2 ca. 39,– DM

Speicher satt!

AMIGA 1000

mit einem Megabyte

Eine weitere Ein-Megabyte-Erweiterung, diesmal aus dem Hause CAS, soll dem Amiga-Speicher auf die Sprünge helfen. Man denke nur daran, daß der Amiga am Anfang seiner Laufbahn mit sage und schreibe nur lächerlichen 256 KByte ausgeliedert wurde, weitere 256 K konnte man sich zwar nachkaufen, aber der Geldbeutel wurde strapaziert. Commodore hatte ein Einsehen und lieferte den Amiga 1000 fortlaufend ohne Aufpreis mit 512 K aus. Damit war das größte Übel beseitigt. Doch schnell war auch dieses Speicherausmaß zu klein, verlangten doch Programme oft mehr als die zur Verfügung stehenden 512 K. DeluxePaint etwa versagte besonders bei höheren Auflösungsstufen aus Speicherplatzgründen seinen Dienst. Der leidlich geplagte A1000-Besitzer schielte unterdessen neidisch auf den Amiga 2000, der serienmäßig mit einem Megabyte an die Kunden übergeht.

Die Zeiten des mangelnden Speichers scheinen vorbei, und die Firma CAS bietet eine preisgünstige Erweiterung an, um mit dem Amiga 2000 gleichzuziehen.

768 K stellen sich vor

Die RAM-Erweiterung wird nicht wie manche anderen auf den Expan-

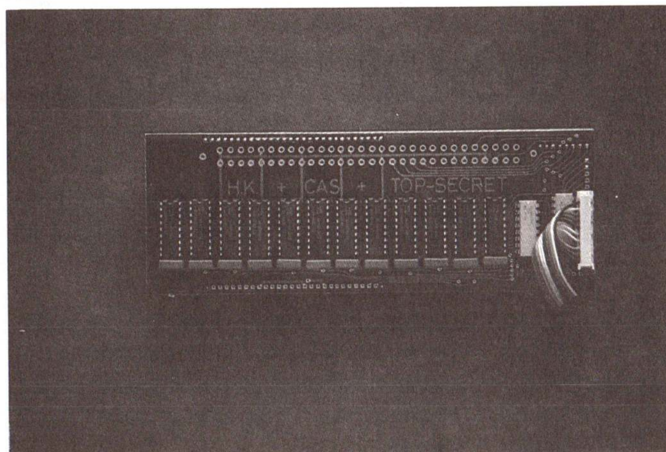


Bild 1: Die 768 KByte-Erweiterung findet unter der Frontplatte des Amiga ihren Platz.

sionsport des Amiga gesteckt, sondert findet am Frontport ihren Platz. Die bereits vorhandene 256 KByte-Karte muß hierbei entfernt werden und kann beiseite gelegt werden. Sie wird nicht mehr benötigt.

Auf der Platine selber finden sich 24 RAM-Bausteine des Typs 41464, wobei sich jeweils 12 auf der doppelseitig bestückten Platine befinden. Bild 1 zeigt die Anordnung der RAMs auf einer Seite der Platine. Diese 24 Chips ergeben zusammen einen Speicher von 768 K und addieren sich mit den bereits im Rechner befindlichen 256 K zu 1024 K. Wer jetzt denkt, es wäre damit getan, die zwei Platinen zu vertauschen, und der Mega-Amiga wäre einsatzfähig, hat falsch gedacht. Es wird zum Lötkolben gebeten.

Zum Lötkolben, bitte

Ein Wort zur Einbauanleitung oder zur Dokumentation. Sie besteht aus einen DIN-A4-Blatt, beidseitig bedruckt, mit einer Abbildung der Hauptplatine des Amiga, in der die betreffenden Chips näher gekennzeichnet sind, einer kurzen 'Schritt für Schritt'-Textanleitung und einer weiteren Skizze, die den Kabelverlauf verdeutlicht. Neben dieser doch sehr dürftigen Dokumentation bekommt man keine weiteren Hilfen erteilt.

Der Lötkolben wird dazu benötigt, noch einige Adresskabel mit verschiedenen Bausteinen auf der Amigaplatine zu verbinden. Da am Frontport die nötigen Adressleitungen nicht vorhanden sind, sind diese Leitungen notwendig. Zusätzlich müssen noch drei weitere Verbindungen auf der Hauptplatine verdrahtet werden. Neben diesen Lötarbeiten ist es notwendig, noch vier IC-Pins mit einem Seitenschneider zu durchtrennen und hochzubiegen. Besitzer eines neueren Amiga 1000 sollten den Garantieverlust bedenken, der beim Öffnen des Rechners entsteht.

Die Lötarbeiten sind nicht so kompliziert, wie es vielleich den Anschein hat. Auch weniger versierte Lötkolbenakrobaten brauchen keine Angst zu zeigen und die Finger von der Sache zu lassen. Innerhalb von einer Stunde ist es ohne weiteres möglich (auch für einen Laien), den Amiga 1000 auf ein Megabyte aufzustocken.

Einen kleinen Zusatz bietet die Erweiterung auch noch. Die Entwickler haben in einer Ecke der Platine zwei Stifte herausgeführt, die durch einen Aufsatz verbunden sind. Durch Herausziehen dieses Aufsatzes verwandelt sich die Erweiterung in eine herkömmliche 256 KByte, die sich in keiner Weise von der von Commodore unterscheidet. Diese Eigenschaft hat folgenden Hintergrund: Einige Programme vertragen sich nicht so recht mit den höher gelegenen Speicherbereichen (Speicheraufbau des Amiga), genauer mit den zusätzlichen 512 K. Meistens stürzt der Rechner bei solch 'fehlerhafter' Software ab. Commdore hat die Probleme der Softwarehäuser erkannt und hat den neueren Workbench-Versionen einen Befehl Namens 'NoFastMem' beigefügt. Dieser schaltet den Speicher über der 512 k-Marke einfach ab und auch wieder zu. Leider ist es bei einigen Programmen nicht möglich, den Befehl aufzurufen, so daß das Programm nutzlos für den Anwender wird. Bei der Erweiterung ist es nun möglich, die beiden Stifte über einen Schalter zu verbinden. Der Schalter kann dann an eine immer erreichbare Position gesetzt werden, so daß man hardwaremäßig den Speicher zu- bzw. abschalten kann.

Fazit

Eine zwar nicht ohne Arbeit verwendbare Speichererweiterung, bei der gelötet und gepetzt werden muß, sie bietet aber durch den Preis einen akzeptalen Kompromiss. Auf den Garantieverfall muß jedoch hingewiesen werden. Man kann sich die Erweiterung auch von einem Fachmann einbauen lassen, was aber zweifellos auf die Geldbörse schlägt. Auch der kluge Gedanke, 512 KByte auszuschalten, kann unliebsame Überraschungen zur Folge haben.

(AK)

+ geringer Platzbedarf
+ Über einen Schalter Speicher ab- und zuschaltbar

– Lötarbeit erforderlich, möglicher Garantieverlust
– knappe Dokumentation

Vertrieb:
CAS Computer
Sprendlinger Landstraße 71
6050 Offenbach

Preis: 378,– DM

Der „Kleine" ganz GROSS
Ein Megabyte im A500

Neben den beiden Speichererweiterungen für den Amiga 1000 haben wir auch eine für den Amiga 500 getestet. Die Erweiterung läßt den 'Kleinen' mit dem Amiga 2000 gleichziehen, in Sachen Speicher versteht sich. Nach erfolreichem Einbau steht dem A500-Anwender ein sattes Megabyte für Programme zur Verfügung.

Die Speichererweiterung stammt aus dem Hause CAS und bietet neben den 512 KByte RAM noch eine akkugepufferte Uhr, so daß die 'Startup-Sequence' (im Ordner 's' auf der

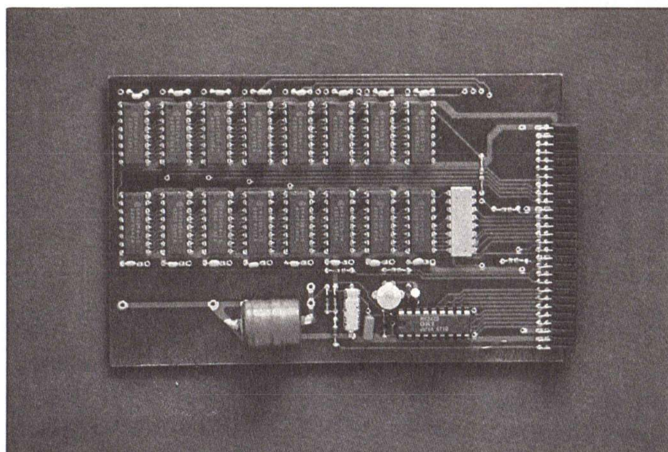


Bild 1: Die 512 KByte-Speichererweiterung mit eingebauter Hardwareuhr

Workbench-Diskette) entsprechend abgeändert werden kann. Damit auch immer die richtige Uhrzeit angezeigt wird, ist eine Änderung notwendig. Im Grunde genommen besitzt die Platine dieselben Hardwareeigenschaften wie die von Commodore angebotene.

Bild 1 zeigt die Speicherweiterung mit den 16 RAM-Bausteinen des Typs 41256, die zusammen die 512 KByte ergeben. Neben den beiden RAM-Reihen ist deutlich der Akku für die Uhr zu erkennen.

Der Einbau

Die 512 KByte finden am Erweiterungsport des Amiga 500 ihren Bestimmungsort. Er ist auf der Unterseite des Rechners und tritt nach Entfernen einer Klappe zutage. Nach Anleitung muß lediglich diese Klappe geöffnet werden, die Platine kann auf die 56polige Leiste der Hauptplatine gesteckt werden, und dem Benutzer steht ein Megabyte zur Verfügung. Kein Lötkolben muß zur Hand genommen oder IC-Pin durchgepetzt werden. Die Garantie verliert demnach nicht ihre Gültigkeit, so daß auch Besitzer von neueren Amigas ohne sich die Speichererweiterung ohne Bedenken zulegen können.

Der Einbauversuch brachte aber einige unliebsamme Überraschungen, von denen die Anleitung natürlich nichts erwähnte. Mit dem schnellen Entfernen der Klappe und dem Aufstecken der Platine war es nichts. Der Grund bestand darin, daß die Dimensionierung des Freiraums für die Erweiterung zu klein oder die Platine zu groß war. Ca. ein Milimeter hinderte uns daran, die Erweiterung erfolgreich einzubauen. Die Ränder der Platine besitzen jedoch genügend Spielraum, so daß an den Seiten ein wenig abgetragen werden kann. Die Feile mußte ausgepackt werden, um damit die Platine zu verkleinern. Vorsicht war geboten, zu leicht konnte man abrutschen und die Platine irreparabel beschädigen. Nach mehreren Ansätzen der Feile hatte die Platine endlich die richtige Größe. Das Aufsetzen stellte sich aber als gar nicht so einfach heraus. Fummelarbeit und festes Drücken brachte die Erweiterung in die gewünschte Position.

Nach Einschalten des Rechners und Laden der Workbench konnte dann ein Megabyte bewundert werden. Auch die Uhr arbeitete einwandfrei, so daß alles in schönster Ordnung war. Die 512 KByte können nur über Software abgeschaltet werden, ein Schalter wurde nicht implementiert. Das Programm 'NoFastMem' ermöglicht ein Ab- und Zuschalten des Speichers. Einige Programme arbeiten nicht mit der Erweiterung zusammen, so daß dieser Befehl manchmal eingesetzt werden muß. Glücklicherweise ist dies die Ausnahme.

Fazit

Für wenig Geld bekommt man seinen Amiga 500 auf ein Megabyte aufgerüstet. Keine Lötarbeiten müssen vorgenommen werden. Der Rechner muß auch nicht geöffnet werden, somit ist kein Garantieverlust zu befürchten. Unsere Probleme beim Einbau sind sicher nicht die Regel, stellten an sich auch kein großes Hindernis dar. Amiga-500-Besitzer können sich diese Speichererweiterung ohne Bedenken zulegen.

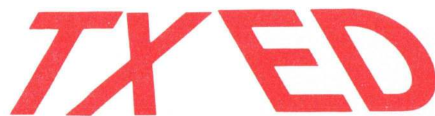
(AK)

\+ Erweiterung mit Hardwareuhr
\+ Einbau ohne Lötarbeit (kein Garantievelust)

– Erweiterung nicht abschaltbar

Vertrieb:
CAS Computer
Sprendlinger Landstraße 71
6050 Offenbach

Preis:
265,– DM mit Uhr
245,– DM ohne Uhr



Die Alternative zu Ed

Editoren sind Werkzeuge zum Erstellen von Quelldateien, also zum Eingeben von Text. Sie werden hauptsächlich zum Programmieren herangezogen. Man kann sie auch ohne weiteres als 'kleine' Textverarbeitungsprogramme bezeichnen, sie besitzen aber nicht deren hohen Standard. Die Schnittstelle zwischen beiden Textsystemen wird immer kleiner werden und bald nur noch durch ein paar spezifische Merkmale erkennbar sein. Der Komfort der Editoren wächst zusehends.

'TxEd' ist ein weiterer Editor zum Erstellen von Quellprogrammen. Er stammt aus dem Hause MICRO-SMITHS INC. und soll eine Alternative zum System Editor 'Ed' des Amiga sein.

Was gibt's fürs Geld?

Mitgeliefert bekommt man eine Diskette, auf der sich der Editor befindet und ein recht dünnes Handbuch, zwanzig läppische Seiten. Die Diskette beinhaltet nicht nur den eigentlichen Editor Version 1.3, sondern noch den Vorläufer – Version 1.2, der auch im Interlace Modus (400 Zeilen) arbeitet, was meiner Meinung nach mit einem 'Standard'-Monitor völliger Unsinn ist, da Augenschmerzen und ähnliche unangenehme Erscheinungen unweigerlich auftreten. Eine Programm namens 'ReTab' wandelt 'Tabs', die mit dem Systemeditor 'Ed' erstellt worden sind, in Leerzeichen um. Neben den Editoren und 'ReTab' findet man noch zwei Dateien zum Konvertieren von Amiga-Texten zu IBM-Texten bzw. umgekehrt. Außerdem existiert noch ein Public-Domain-Programm, das eine Reihe von Textformatierungs-Funktionen beinhaltet. Das Programm arbeitet aber nicht direkt mit 'TxEd' zusammen. Die erstellten Texte können nur nachformatiert werden.

Das bietet TxEd!

Der Editor kann von der Workbench sowie vom CLI aus gestartet werden, wobei beim ersteren Laden die 'Word Wrap'-Funktion (Wortumbruch) angewählt ist, beim Starten vom CLI aus nicht. Nach erfolgreichem Laden erscheint ein Fenster, bei dem in der Kopfzeile der Name der Datei und eine Anzeige der Zeilen- und Spaltennummer steht. Ebenfalls ist auffallend, daß der ganze PAL-Bildschirm (256 Zeilen) sofort ausgenutzt wird. Das Fenster läßt sich über herkömmliche System-Gadgets in den Hinter- oder Vordergrund setzen sowie verkleinern oder vergrößern. Nach Drücken der rechten Maustaste taucht die Menueleiste am oberen Bildschirmrand auf und legt sechs verschiedene Menues zur weiteren Auswahl dar. Die Menuepunkte können dann wie bekannt angewählt werden. Die wichtigsten Punkte sind zusätzlich noch über Tastatur-Sequenzen zu erreichen – was natürlich Freunde der konservativen Methode freuen wird. Die Benutzung der Tastaturcodes geht nach Einarbeitung in den Editor wesentlich schneller vonstatten als das Arbeiten mit der Maus. Bild 1 zeigt eine Menue mit den dazugehörigen Tastatur-Sequenzen. Der Quelltext kann sofort eingegeben werden, keine Parameter müssen übergeben oder eingestellt werden. Am sinnvollsten ist es, 'TxEd' auf die System-Diskette in den Ordner 'c' zu kopieren, da der Editor von dort jederzeit direkt aufgerufen werden kann. Doch was bietet 'TxEd' einem Programmierer an nützlichen Utilities?

Die Features

Das Anwählen des erstes Menues legt eine ganze Reihe von Optionen frei. 'Clear' zum Löschen des Puffers oder 'Save' und 'Save as' zum Speichern des Textes mit neuen Namen bzw. ohne Änderung. Das Laden einer Datei geschieht ebenfalls im ersten Menue. Eine sogenannte 'File-Selektor Box' wird hierzu herangezogen. Bild 2 zeigt eine solche Box. Jedoch muß vor dem Laden eines zweiten Textes gewarnt werden, 'TxEd' überschreibt hemmungslos den alten. Keine zusätzliche Warnung wird ausgegeben. Neben diesen Optionen existieren noch 'Print Clip' zum Ausdrucken des Textes sowie 'Quit' zum Verlassen des Editors. Sie sollten sich immer vergewiessern, ob

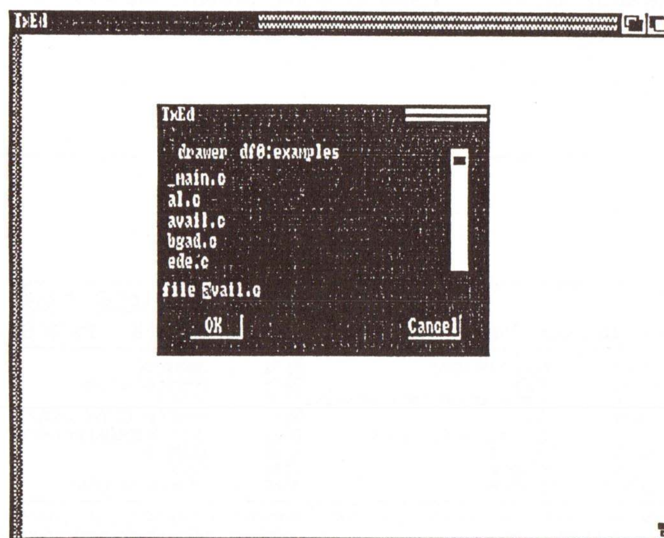


Bild 1:
TxEd gestattet es, die Menuepunkte auch über Tastatur- Sequenzen aufzurufen.

der Text abgespeichert wurde, auch hier gibt der Editor keine Warnung aus, sondern beendet sofort das Editieren. Zusätzlich zeigt dieses Menue noch den momentan vorhandenen freien Speicher an. Leider gibt der Editor nur 35 000 Bytes zum Erstellen von Texten frei. Im Handbuch konnte ich keinen Hinweis finden, ob es möglich ist, diesen Speicher zu vergrößern. Die rechte Seite des Menuebalkens zeigt immer eine dazugehörige Tastatur-Sequenz auf.

Das zweite Menue beinhaltet verschiedene Editiermöglickeiten. Angefangen von der Blockmarkierung, die sehr einfach und komfortabel mit der Maus oder dem Cursor vollzogen werden kann, über Blockbefehle bis hin zu verschiedenen Lösch-Optionen. Ein Puffer speichert den zuletzt gelöschen Text, der nach Bedarf wieder zurückgeholt werden kann.

Menue Nummer drei befaßt sich mit der Cursorsteuerung. Es kann zum Anfang bzw. zum Ende des Textes, in eine bestimmte Zeile, ein Wort zurück bzw. vorwärts oder den Anfang oder Ende einer Zeile gesprungen werden. Speziell die Cursorsteuerung ist über Tastatur-Sequenzen wesentlich schneller als mit der Maus.

Im 'Search'-Menue kann veranlaßt werden, nach einer bestimmten Zeichenkette zu suchen oder Zeichen zu vertauschen. Die Optionen können sowohl rückwärts wie auch vorwärts arbeiten. Ebenfalls kann eingestellt werden, ob Groß-, Kleinschreibung beachtet werden soll. Der Benutzung von Wildcards oder Jokern steht ebenfalls nichts im Wege. Bei 'Replace' stehen dem Anwender vor jedem Vertauschen drei Möglichkeiten zur Wahl. Zum ersten das Abbrechen der Option, zum zweiten das Tauschen des Textes und zum dritten das Tauschen im ganzen Text ohne vorhergehende Abfrage. Die Optionen erledigen ihre Arbeit sehr schnell, kein unnützes Warten liegt an.

Das 'Random'-Menue

Das letzte Menue bietet seine Besonderheiten, deshalb möchte ich ihm etwas mehr Beachtung schenken. Die Option 'Hex Entry' erlaubt es, in der oberen Fensterzeile ASCII-Werte einzugeben, die nach Drücken der Return-Taste das entsprechende Zeichen an der Cursorposition ausgeben. Es

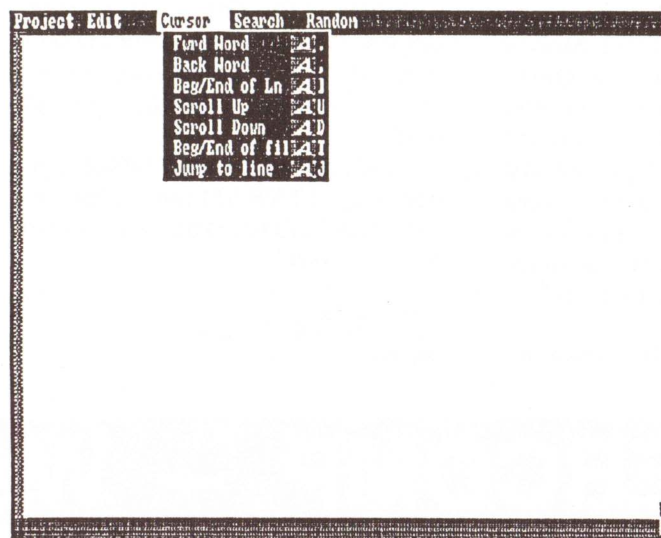


Bild 2: Das Laden einer Datei über die 'File-Selektor-Box' ist schnell und komfortabel.

müssen jedoch dezimale Werte eingesetzt werden, hexadezimale Werte erkennt 'TxEd' nicht an.

Mit dem zweiten Menuepunkt kann die Bildschirmfarbe geändert werden, nach jedem Aufrufen des Punktes erscheint eine neue Farbe auf den Bildschirm. Jedoch stehen nur vier Farben zur Verfügung. Durch Ändern der Farbschieber im Programm Preferences können vier andere Farben definiert werden.

Der dritte Menuepunkt des 'Random'-Menues gestattet es dem Anwender, das CLI aufzurufen. Mit dieser Option kann der Programmierer sich zwischendurch anderen Aufgaben widmen, ohne den Editor abzubrechen.

Mit 'More TxEd' wird ein weiterer Editor gestartet. Die Funktionen 'Cut Block' und 'Insert Block' können dazu herangezogen werden, ganze Blöcke von einem Editor in den anderen zu verschieben.

Die Funktion 'Strip Etc' kann verwendet werden, um IBM-PC Dateien in Amiga Dateien zu konvertieren. Die Funktion entfernt <CR> und andere CTRL-Sequenzen vom Text-Puffer. Nützlich, bedenkt man, das Quelldateien vom IBM-PC Standard ohne weiteres auf dem Amiga Verwendung finden können.

Eine sehr ungewöhnliche Funktion für einen Editor nennt sich 'Word Wrap'. Sie veranlaßt den sofortigen Wortumbruch am Ende einer Zeile. Je nach Einstellung kann das nach 60 bzw. 80 Zeichen geschehen. Ist die Funktion nicht aktiviert, stehen dem Benutzer 256 Zeichen pro Zeile zur Eingabe zur Verfügung. Für einen Editor ist 'Word Wrap' natürlich nicht von immenser Bedeutung, besitzt aber trotzdem seinen Nutzen. Hilfreich ist auch das Kennzeichnen des Punktes mit einen Haken, jenachdem ob angewählt oder nicht.

'Overstrike' ist der letzte Menuepunkt des 'Random'-Menues. Das Anwählen wird ebenfalls, wie bei der 'Word Wrap' Funktion, mit einem Haken gekennzeichnet. Ein Überschreiben alter Zeichen ist möglich, der 'Einfüge'-Modus wird durch Anwählen des Punktes ausgeschaltet.

Fazit

Ohne Zweifel ist 'TxEd' wesentlich komfortabler und flexibler als der Systemeditor 'Ed'. Für Programmierer ist er ein vorzügliches Werkzeug zum Schreiben von Programmen. Mängel, die 'Ed' aufweisen mußte, wurden beseitigt. Die doppelte Belegung der Menuepunkte mit Tastatur-Sequenzen ist sehr nützlich, denn sie läßt nach Einarbeitung in den Editor ein schnelleres Anwählen der verschiedenen Funktionen zu. Ebenfalls sehr positiv zu bewerten ist die schnelle Abarbeitung von Suchroutinen oder Scrolloptionen. Auch das Einlesen von Verzeichnissen mit Hilfe der 'File-Selektor-Box' geht sehr zügig vonstatten, allzulange Wartezeiten wie bei manchen anderen Programmen gehören der Vergangenheit an. Negativ zu bewerten ist lediglich, daß keine Option gefunden werden konnte, die den Arbeitsspeicher des Editors heraufsetzt. 35 000 KByte scheinen zwar auf den ersten Blick

ausreichend, für größere Programme ist der Speicher aber meiner Ansicht nach zu gering. Das dünne englischsprache Handbuch ist ein weiterer Kritikpunkt, der aber nicht stark ins Gewicht fällt, da das Programm fast selbsterläuternd ist. Ein Kopierschutz existiert ebenfalls nicht, was für einen Editor, der zum Programmieren verwendet werden soll, auch sachgemäß ist.

Der Preis von 128 DM erscheint angemessen für einen leistungsfähigen Editor, bedenkt man nur die sehr zahlreichen Funktionen, die von einem Editor nicht unbedingt verlangt werden.

Lauffähig auf Amiga 1000 (Kickstart 1.2) 512 KRAM und höher, sowie ohne Einschränkung auf Amiga 500 und 2000.

(AK)

Bezugsadresse:
DTM, Wiesbaden, IM 069/707 11 02
DM 128,–

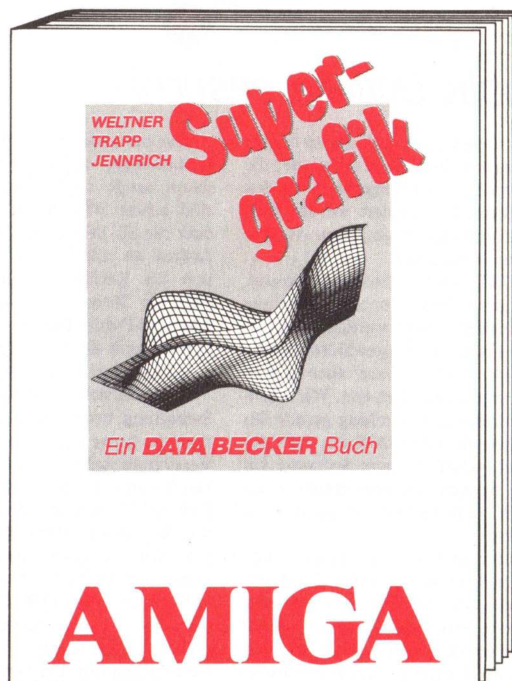
+ Steuerung mit der Maus und Tastatur-Sequenzen
+ Konvertierung von Dateien möglich
+ Leichtes Erlernen
+ Schnelle Optionen
+ Kein Kopierschutz

– Keine Option zum vergrößern des Arbeitsspeichers
– Dünnes englisches Handbuch

Der Amiga setzt Grafikstandards.

Dieses Buch auch:

Es gibt viele Rechner, aber nur einen Amiga. Hier hält die Werbung wirklich das, was sie verspricht: „Unschlagbar in der Kombination Text und Grafik/Farbe.“ Eine Reihe Bücher widmet sich daher speziell diesem Thema; aber sicherlich keines so ausführlich wie Amiga Supergrafik. Denn hier finden Sie tatsächlich das gesamte Know-how zu Grafik und Animation auf dem Amiga – Wissen, das Sie für eigene, eindrucksstarke Programme nutzen können. Wie lassen sich die phantastischen Möglichkeiten dieser Grafikmaschine bis zum Letzten ausreizen? Wie erreicht man die Grenze des Machbaren? Dieses Buch verrät Ihnen die Antwort. Wer sich hier soweit vorwagt, wird in Dimensionen stoßen, die er selbst einem Amiga nicht zutrauen würde. Aber keine Sorge: Amiga Supergrafik wurde nicht nur für Profis geschrieben. Es bietet jedem etwas. So gibt es zahlreiche Einsteiger-Programme, die das nötige Grundwissen vermitteln, ebenso wie Programme für den fortgeschrittenen BASIC-Programmierer. Profis hingegen erfahren, wie sie die Grafik von C aus ansprechen können. Kurzum ein Buch, das Ihnen genau das Know-how vermittelt, das Sie brauchen: Grafikprogrammierung mit den vorhandenen BASIC-Befehlen, Nutzung der Libraries, die Register der Grafik-Chips, Aufbau und Programmierung von Screens, Windows, HAM, Halfbrites und Interlaces aus BASIC und C, 1024 x 1024 Punkte Superbitmap, gepufferte Multitasking-Hardcopy-Routine – zum Thema Grafik werden Sie in Amiga Supergrafik nichts vermissen.



Amiga Supergrafik
Hardcover, 686 Seiten, DM 59,–

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort

PINB

Autoren gesucht

Die KICKSTART-Redaktion sucht Programmierer, die sich mit dem AMIGA auskennen und darüber zu berichten wissen, sei es über Soft- oder Hardware.

Wenn Sie eigene Programme geschrieben haben, die Sie gerne veröffentlichen würden, so lassen Sie sie nicht in der Schublade verschwinden, sondern schicken sie uns!

Neben Programmlistings suchen wir auch Tips & Tricks über den AMIGA. Angefangen bei Assembler bis hin zu Basic, C, Pascal oder anderen Sprachen. Auch die Bastler sind aufgerufen, über ihre Fachgebiete (z.B. Floppies, Drucker, Harddisk, Monitor) zu berichten. Das Honorar erfolgt nach Vereinbarung.

Über Ihre Beteiligung würden wir uns sehr freuen.

KICKSTART-Redaktion
'Programmeinsendung'
Industriestr. 26
6236 Eschborn
Tel. 06196/41245

Public Domain gesucht

Für viele ist Public-Domain bereits ein Begriff. Es sind Programme, die von den Autoren freigegeben wurden und somit frei kopiert werden dürfen. Damit stehen sie jedem Anwender zur Verfügung.

KICKSTART hat sich entschlossen, eine eigene Public-Domain-Sammlung zu schaffen. Darin werden neben einer Auswahl an ausgewählter, schon existierender Software auch eigene Programme enthalten sein. Wir möchten mit dieser Sammlung gerade für Neulinge der AMIGA-Szene die Möglichkeit schaffen, gute Software für ihren Rechner zu bekommen, ohne tief in den Geldbeutel greifen zu müssen.

Schon jetzt hat sich gezeigt, daß die Qualität der privaten Programmierkunst an manches professionelle Produkt heranreicht und sich davor nicht verstecken muß. Zwar ist meist nicht die Leistung einer Softwarefirma zu erreichen, doch bestätigen Ausnahmen bekanntlich die Regel.

Nicht nur große Programme sind dabei gefragt; auch Utilities, die im Einzelfall von unschätzbarem Wert sein können, sind wichtige Bestandteile der Public-Domain. Auch ausgefeilte Routinen und Funktionsbibliotheken können für andere Programmierer sehr wichtig sein. Insgesamt sind dem Einfallsreichtum keine Grenzen gesetzt.

Was bei anderen Rechnersystemen funktioniert, sollte beim AMIGA schon lange funktionieren – was sind schon 500 MS-DOS Disketten oder die ST-Sammlung! Deshalb der Aufruf an alle Programmierer, die sich des Rechners mächtig fühlen: Tun Sie Ihren Teil dazu, um die AMIGA-Public-Domain Szene anzukurbeln und sie in der Computerwelt zu einem Begriff zu machen.

Wenn Sie sich an der KICKSTART-Sammlung beteiligen und Ihre Programme gerne der Allgemeinheit zur Verfügung stellen möchten, so schikken Sie uns das/die Programm/e einfach auf Diskette zu. Bitte bestätigen Sie, daß das Programm von Ihnen geschrieben wurde und daran keinerlei kommerzielle Rechte bestehen.

Wir halten die Public-Domain Software für eine sehr nützliche Kreation und möchten unseren Teil zu ihrer Verbreitung tun. Die Sammlung wird ab sofort zum Selbstkostenpreis über die Redaktion zu erhalten sein.

KICKSTART Redaktion
'PD-Einsendung'
Industriestr. 26
6236 Eschborn
Tel. 06196/41245

Tips & Tricks

Ihre Erfahrungen

Der AMIGA ist schon ein toller Rechner, aber nicht immer kommt man mit ihm problemlos zurande. Manchmal hat man seine Qual um endlich das Gewollte zu erreichen. Haben Sie es nicht auch schon erlebt, daß man an einem Problem oft mehrere Stunden, wenn nicht sogar Tage herumbastelt? Deshalb sollen die Lösungen hier zur Sprache kommen.

Schicken Sie uns doch einfach Ihre Programmiertips zu oder die Tricks zu Anwenderprogrammen aller Art. Natürlich werden alle Veröffentlichungen honoriert!

MERLIN-Computer GmbH
KICKSTART-Redaktion
'Tips & Tricks'
Industriestr. 26
D-6236 Eschborn

O A R D

Ihre Wertung bitte!

Meine Meinung zu...

Diese Sparte ist für Ihre persönliche Meinung freigehalten. Wenn Sie über ein AMIGA-Produkt besonders glücklich oder unglücklich sind, dann teilen Sie es uns mit. Wir werden Sie mit denen anderer Leser veröffentlichen um somit der öffentlichen Meinung Sorge zu tragen.

Sind Sie mit einem Produkt zufrieden oder könnten Sie klagen weil es fehlerhaft und unausgereift ist?

Hatten Sie mit Hardwareerweiterungen Probleme oder funktionierten Sie einwandfrei?

Wie steht es mit der AMIGA-Literatur. Sind Sie mit bestimmten Büchern zufrieden, können Sie eines besonders empfehlen?

Zu all diesen oder auch anderen Themen, dürfen Sie sich äußern. Sie werden dann im Rahmen der Möglichkeit (Länge) veröffentlicht.

Zuschriften an:
MERLIN-Computer GmbH
KICKSTART-Redaktion
'Meine Meinung'
Industriestr. 26
D-6236 Eschborn

MERLIN-Computer GmbH
KICKSTART-Redaktion
'Programmeinsendung'
Industriestr. 26
D-6236 Eschborn
Tel. 0 61 96 / 48 18 11

MERLIN-Computer GmbH
KICKSTART-Redaktion
'PD-Einsendung'
Industriestr. 26
D-6236 Eschborn
Tel. 0 61 96 / 48 18 11

MERLIN-Computer GmbH
KICKSTART-Redaktion
'Comic'
Industriestr. 26
D-6236 Eschborn
Tel. 0 61 96 / 48 18 11

MERLIN-Computer GmbH
KICKSTART-Redaktion
'Clubinfo'
Industriestr. 26
D-6236 Eschborn
Tel. 0 61 96 / 48 18 11

Wir gratulieren

Wir bedanken uns vielmals bei allen Lesern, die sich an der Umfrage beteiligten. Aus den Einsendungen wurden fünf Abos, der KICKSTART (siehe erste Umschlagseite) und zusätzlich fünfmal das Spiel 'THE FAERY TALE ADVENTURE' verlost.

Wir bedanken uns an dieser Stelle bei der Firma Micro Illusions für die Bereitstellung der Programme.

Die Sieger/innen:

Ein KICKSTART Jahresabo erhalten:

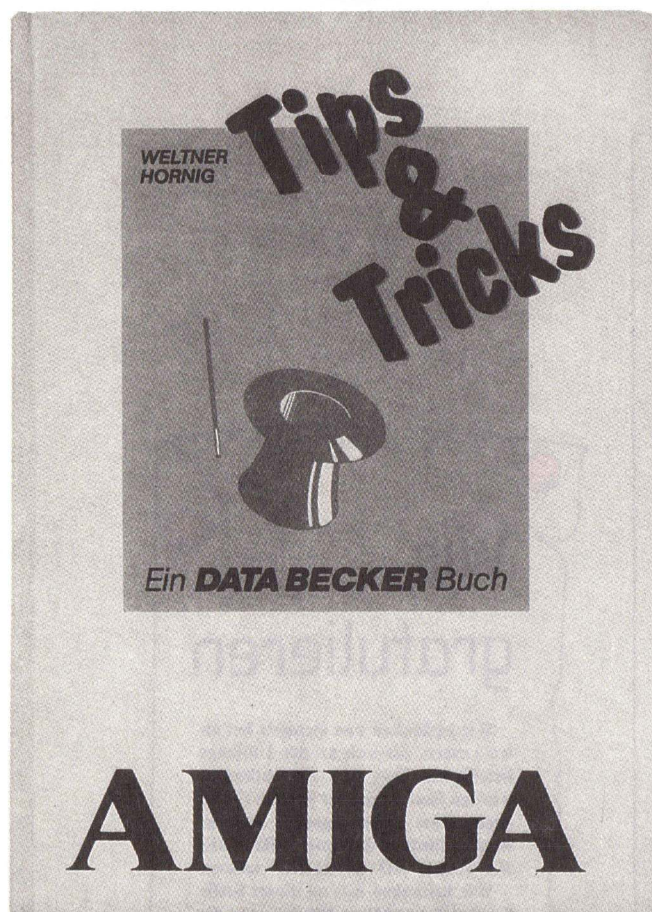
- H. Alphons / Rotterdam
- I. Koschollek / Hamburg
- R. Studtrucker / Wetter
- M. Stosick / Gröbenzell
- J. Pet ermann / Frankfurt

Je ein THE FAERY TALE ADVENTURE erhalten:

- S. Tandecki / Schöppenstedt
- D. Kononka / Lohmar
- K.-M. Friedrich / Hannover
- U. Menze / Oldenburg
- C. Rohm / Markt Schwaben

Herzlichen Glückwunsch!

DAS AMIGA-BASIC BUCH



Fast in jedem erwacht einmal der Wunsch, seinen Amiga selbst zu programmieren, und bei der Wahl der Sprache wird oft auf den mitgelieferten Basicinterpreter zurückgegriffen. die Vielzahl der Befehle sollte eigentlich einiges ermöglichen, doch das typische Commodorehandbuch schreckt die meisten wieder ab. Eine Alternative stellt das „Amiga Basic" Buch von Data Becker dar. Dieses 775 Seiten starke Werk bietet dem Anfänger eine wichtige Starthilfe. Die locker geschriebene Einleitung bereitet den Leser bereits darauf vor, daß das nun folgende in einer unterhaltsamen, lockeren und trotzdem lehrreichen Art geschrieben wurde. Es wird eine Basiceinführung erbracht, die zum Experimentieren anregt, was gerade für den Anfänger von großer Wichtigkeit ist. Im folgenden werden insgesamt 6 verschiedene Programme besprochen, unter ihnen ein Malprogramm und eine Dateiverwaltung, sowie ein Programm zur Erstellung von Videotiteln, welches auch in der Lage ist, IFF-Bilder in Basic einzulesen. Anfangs geschieht die Besprechung in kleinen Portionen, die mit der Zeit immer größer werden, bis man sich zum Schluß auf die neuen Befehle beschränkt. Die anfangs sehr ausführlichen Erklärungen ermöglichen auch dem Anfänger das Verstehen der größeren Programme und sind eine sehr gute Grundlage für eigene Projekte. In dem Buch werden sowohl die Grafik-, und Ton- als auch die Dateibefehle behandelt, wobei der Sonderteil etwas zu kurz gekommen ist (lediglich 46 Seiten). An den Stellen, wo die Autoren es für nötig hielten, wurden Ausflüge in die Welt der Bits und Bytes, der Zahlensysteme oder in die Mathematik unternommen, alles jedoch in einem Rahmen, der den Leser nicht abschrecken sollte. Zum Schluß ist der Leser dann in der Lage, Befehle selber zusammenzubauen oder ein eigenes Druckerprogramm zu erstellen. Der 200 Seiten starke Anhang enthält eine genaue Erklärung der Fehlermeldungen, sowie der Basicbefehle und ein kleines Fachwörterlexikon. Hierdurch ist das Buch auch als Nachschlagewerk zu gebrauchen.

Fazit:

Ein Buch, das auf keinen Fall für den Fortgeschrittenen gedacht ist, dem Anfänger jedoch einen guten Start in die Welt der Basicprogrammierung bietet und das seinen etwas hohen Preis (59,– DM) wert ist.

TIPS & TRICKS! AMIGA

Von DATA BECKER kommt ein weiteres Buch mit dem Titel „Tips & Tricks", diesmal für den Amiga. Die Autoren Ralf Hornig und Tobias Weltner liefern jedoch nicht, wie zu erwarten, allgemeine Ratschläge, sie zeigen vielmehr, wie man bestimmte Probleme in AmigaBASIC oder C löst. Entsprechend hat das Buch in zwei größere Kapitel: Programmiertips einmal in BASIC, einmal in C. Allerdings gibt die Einleitung auch Antworten auf Fragen, die ein Amiga-Anfänger vielleicht stellen könnte. Der AmigaBASIC-Teil bringt eine Fülle von Tips und Tricks für das Programmieren. Auch fortgeschrittene Programmierer können sich hier nützliche Informationen holen. In diesem Abschnitt werden unter anderem die Programmierung von verschiedenen Druckmodi oder das Zurückgreifen in AmigaBASIC auf Systemroutinen behandelt; speziell werden hier Routinen der Intuition-Library genutzt, aber auch andere Library-Routinen finden Beachtung. Alle Themen werden in einem Programmbeispiel näher erläutert, so daß der Anwender die Ausführung nach dem Abtippen bewundern kann. Die Programmbeispiele, die von uns eingegeben wurden, waren sämtlich lauffähig. Leider ist dies nicht immer der Fall!

Das zweite Kapitel des Buches widmet sich ausschließlich der C-Programmierung. Zu Beginn werden einige allgemeine Hinweise zur Programmierung in C gegeben, anschließend erklären die Autoren mit kleinen Programmbeispielen die verschiedensten Problemlösungen. Unter anderem wird auf verschiedene Datei-Utilities oder auf die Programmierung von Fenstern und Bildschirmen eingegangen.

Insgesamt betrachtet liefert das Buch eine recht gute Sammlung von Programmbeispielen, die auch für fortgeschrittene Programmierer von Nutzen sind. Natürlich kann es einige Probleme nur am Rande erwähnen: Für manchen Anwender könnten die Schwierigkeiten erst anfangen. Als Nachschlagewerk ist es deshalb nicht zu empfehlen. Programmieranfänger sowie Fortgeschrittene, vor allem in BASIC, können sich den Band durchaus zulegen.

(AK)

Ralf Hornig, Tobias Weltner
Tips & Tricks: AMIGA
DATA BECKER Verlag
2. Auflage 1987
364 Seiten / 49,– DM.



WENN DER BEIWAGEN ZUM ÜBERHOLEN ANSETZT...

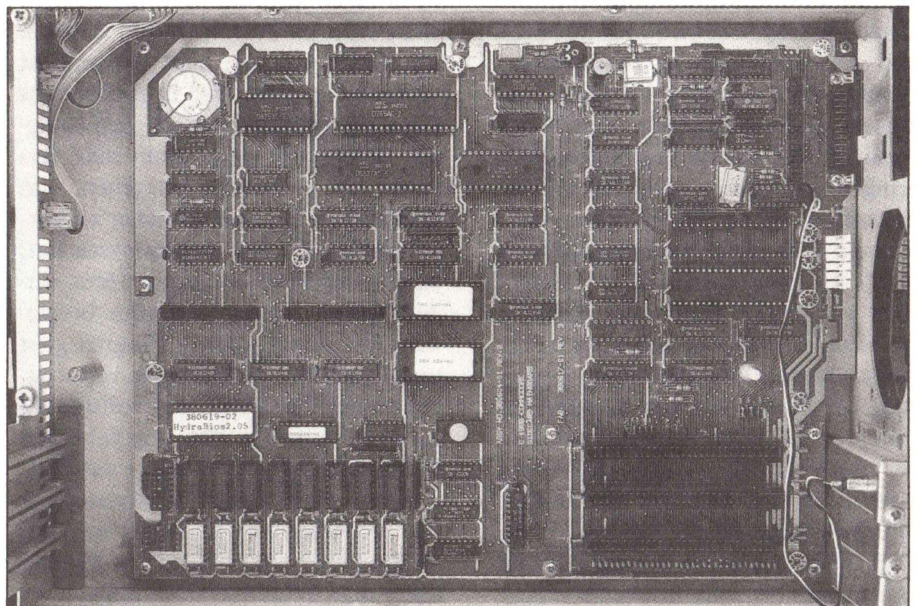
...ist es beim Motorradfahren meist zu spät. Beim AMIGA-SIDECAR-Gespann wird es dann aber erst interessant. Zwar wird es der PC-Teil auch nach den in diesem Artikel beschriebenen Tuning-Arbeiten nicht fertigbringen, den AMIGA auf die hinteren Ränge zu verweisen, aber immerhin.

Hier wird es jedoch nicht nur um Geschwindigkeit gehen. Vielmehr werden einige Änderungen Erweiterungen der Hardware vorgestellt, die das Leben und Arbeiten mit dem SIDECAR schöner und leichter machen.

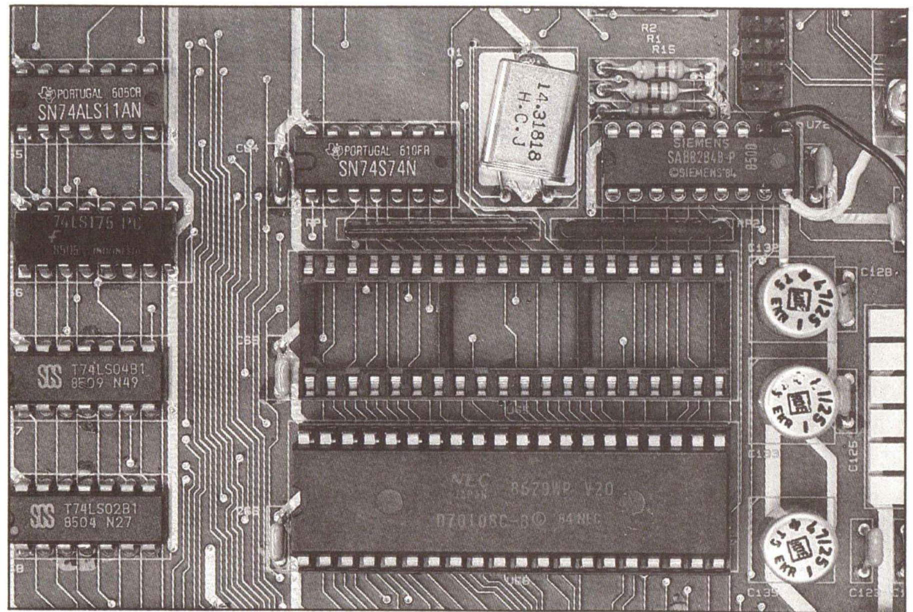
Nur zwei Tage, nachdem der SIDECAR des Autors das Neonlicht der Schreibtischlampe erblickt hatte, mußte er die erste Operation über sich ergehen lassen. Die Diagnose lautete: Akute Mangelversorgung mit RAM. Mit dem Standardausbau von 256 kByte war kein Staat zu machen. WORDSTAR und TURBO-PASCAL geben sich wohl noch mit diesen beengten Arbeitsverhältnissen zufrieden. Aber wehe dem, der neuere Programme hat. Heutzutage werden die Programme meist so speicherplatzintensiv, daß eine Aufstockung der Kapazität fast unumgänglich ist. Wenn man da an früher denkt: CP/M, 64 kByte – und es funktionierte trotzdem. Zu den Glanzzeiten von Z80 und 6502 hätte man sich für 256 kByte auch einen Urlaub auf den Malediven leisten können. Heute ist ein Tag Schwarzwald (aber bitte ohne Klinik!) schon teurer.

Für die Aufrüstung auf 512 kByte braucht man nichts weiter als acht RAM-Chips 256 kBit. Um höhere Speicherkapazitäten zu erlangen, muß man schon tiefer in die Tasche greifen und sich eine Karte für einen Slot besorgen. Wer das erwägt, sollte gleich Nägel mit Köpfen machen und sich eine Multifunktionskarte kaufen. Diese gibt es von den meisten Herstellern in zwei Ausführungen: die einfache stellt „nur" eine batteriegepufferte Uhr, Gameport, serielle und parallele Schnittstelle zur Verfügung, die größere bietet zusätzlich noch 384 kByte RAM. Die RAM-bestückte Multifunktionskarte bietet so in nur einem belegten Slot Nützliches zu einem meist annehmbaren Preis. Man sollte aber auf einen Probelauf in einem SIDECAR bestehen oder sich ein Umtausch-Recht einräumen lassen, da es mit der Karte eventuell Probleme geben könnte, z. B. wenn die parallele Schnittstelle nur als LPT1: betrieben werden kann. Diese kommt sich dann mit der Emulation auf der AMIGA-Seite ins Gehege. Doch nun zu jenen, die nur auf Speicher Wert legen und denen 512 kByte ausreichen. Wie erwähnt, benötigt man nur 8 ICs. Die verwendbaren Bausteine tragen die Bezeichnung 41256. Dabei handelt es sich um dynamische RAMs. Sie sind Bit-orientiert ausgelegt (im AMIGA findet man 41464-Chips, die 4-Bit-orientiert sind, aber auch 256 kBit – 4x64k – haben). Der Hersteller ist unwichtig, da der SIDECAR alles verträgt, was ihr in die Sockel kommt. Die Zugriffszeit sollte 150ns nicht übersteigen, was in der Praxis immer erfüllt ist. Meine Lieferanten hatten gar keine langsameren 41256-Chips im Programm. Gesamtkosten: etwa 60 DM.

Der Einbau ist recht einfach. Man zieht den Netzstecker ab und trennt



dann AMIGA und SIDECAR voneinander. Links und rechts sitzen je zwei Kreuzschlitzschrauben, die entfernt werden müssen. Jetzt kann man die Haube ein Stück nach vorne ziehen und nach oben abheben. Wer auch schon vor Begeisterung abgehoben ist, hat wahrscheinlich die Anschlußdrähte der Betriebs-LED mit Schwung abgerissen (Erfahrungswerte, alles Erfahrungswerte...). Wenn man nun in das Gerät blickt, sieht man rechts neben dem Floppy-Laufwerk eine Etage tiefer acht leere IC-Sockel. In diese müssen die Speicher gesteckt werden. Man sollte darauf achten, daß kein Bein abbricht oder außen am Kontakt vorbeigesteckt wird. Die Kerbe (oder Farbpunkt oder Vertiefung) muß dabei nach links zeigen (wie bei der Reihe nebenan). Der Rechner muß nun nur noch erfahren, daß er mehr Speicher ab jetzt hat. Dies bewerkstelligt man mit den DIP-Schaltern, die vor den Speicherbänken zu finden sind. Auf dem achtfachen Mäuseklavier wird Nummer 3 auf On und Nr. 4 auf Off geschaltet. Et voil: 512 kByte Arbeitsspeicher.



Tritt aufs Gas

Wenn es bisher nur um „Hubraum“ ging, wird jetzt richtig getunt. Und es wird auch nicht lange gezögert, man geht gleich in die Vollen und tauscht den Prozessor. Mancher wird jetzt erschrecken, an solche Aktionen beim Amiga denken und sagen: Zu teuer und zu kompliziert. Mitnichten, die Kosten betragen etwa 20 bis 30 Mark und der Austausch ist kinderleicht. Denn es wird wirklich nur ein IC getauscht und sonst nichts. Das Wunderding, was solche Großtaten vollbringen soll, nennt sich PD 70108. Bekannter ist es jedoch unter dem Namen V20. Dieser Prozessor von NEC ist pinkompatibel zu einem 8088 von Intel, kennt alle Befehle, die dieser verarbeitet und noch einige mehr. Durch eine geschicktere Auslegung des Prozessors war es möglich, diesen sogar noch schneller werden zu lassen als das Original. Wobei man nicht von Original und Kopie sprechen kann, da die Realisation im Chip selbst gänzlich verschieden ist (soweit dies bei einer konventionellen CPU möglich ist). So besitzt der V20 intern zwei Datenbusse, wo der Intel nur einen hat. Durch die jetzt mögliche gleichzeitige Verschiebung ist schon ein erheblicher Geschwindigkeitsvorteil gegeben. Eine zweite Besonderheit, die in diesem Zusammenhang nicht so wichtig ist, besteht darin, daß der V20 noch einen 8080 implementiert hat. So bietet sich die Möglichkeit, alte CP/M-Software auch auf dem PC-Kompatiblen laufen zu lassen. Doch ist sich der Programmierer hier selbst überlassen, da mir kein Programm bekannt ist, das dies unterstützt.

Der Hauptgrund, einen V20 zu benutzen, wird der Geschwindigkeitszuwachs sein. In welchen Größenordnungen sich der Gewinn bewegt, sieht man in Tabelle 1. Ein Gewinn im Mittel von ca. acht Prozent ist schon recht ordentlich in Anbetracht der aufgewandten Mittel. Auf anderen Rechnern muß man teils das fünf- bis zehnfache ausgeben, um drei bis vier Prozent schneller zu werden. Wer sich jetzt entschlossen hat, seinem Rechner ein neues Herz einzupflanzen, sollte folgendes bedenken: Mit einem solchen Eingriff geht unweigerlich jede Garantie verloren. Man muß das Gerät fast vollständig zerlegen, was für den unerfahrenen Benutzer eventuell Probleme aufwirft.

Auch sind einige Vorkehrungen zur Sicherheit des Rechners und des Heimwerkers zu treffen. Der Mechaniker mag keine Netzspanunngen ertragen, die durch nicht abgezogene Netzkabel noch im Gerät anliegen. Der SIDECAR kann da dem Techniker gefährlicher werden. Durch unglückliche Teppichboden-Schuhsohlen-Kombinationen kann man sich dermaßen elektrostatisch aufladen, daß diese Spannung schon ein IC zerstören kann. Man sollte alle Metallteile, mit denen man im Rechner hantiert, vorher kurz erden. Das gleiche gilt für die Finger.

Wer jetzt immer noch nicht abgeschreckt ist, kann den Kreuzschlitz-Schraubendreher in die Hand nehmen und das Gerät öffnen. Vorgehensweise: siehe oben. Hat man nun das Ding offen vor sich stehen, muß als nächstes das Chassis mit Netzteil und Floppy-Laufwerk entfernt werden. Dazu werden die beiden Schrau-

	8088	V20
BM 1	38.88 sec	37.40 sec
BM 2	58.49 sec	46.02 sec
BM 3	28.23 sec	28.34 sec
BM 4	50.64 sec	47.94 sec
BM 5	25.48 sec	24.77 sec
BM 6	39.60 sec	37.78 sec

Die Benchmarks wurden in TurboPascal geschrieben und liefen als COM-Files. In den Programmen wurde folgendes ausgeführt:

BM 1: Anlegen und Füllen eines Arrays
BM 2: Ausgabe auf dem Bildsch. mittels WRITELN
BM 3: Delay-Befehl (erzeugt Wartezeit)
BM 4: Mathemathische Grundoperationen
BM 5: Inline-Maschinencode Indirekte Adressierung
BM 6: Inline-Maschinencode POP / PUSH

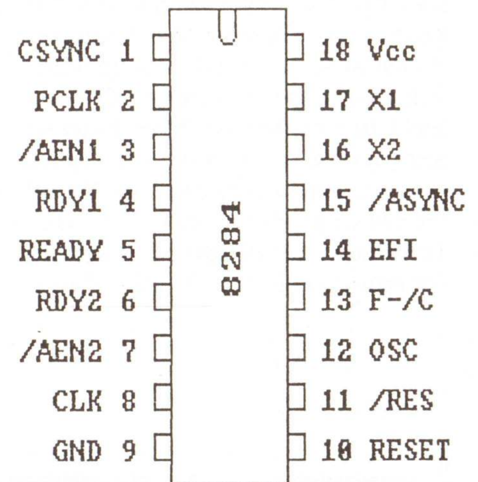
ben auf der Rückwand und zwei vorne links und rechts neben der Floppy gelöst. Am Netzteil sind einige Kabelverbindungen zu lösen, damit man das Chassis ganz abnehmen kann, sodaß es nicht mehr stört. Da der Stecker auf dem Motherboard (Hauptplatine) „gut" zugänglich unter dem Ventilator angebracht ist, ist es einfacher, diesen zuerst zu entfernen, um dann den Stecker ohne Fingermikado abziehen zu können. Nachdem das Floppy-Bus-Kabel abgezogen ist, kann man das Teil ganz entfernen und sicher lagern. Nächster Punkt ist die Interface-Platine. Auch hier müssen zuerst die Kabelverbindungen gelöst werden, um dann die Platine abnehmen zu können. Wenn man jetzt das gute Stück in zuvor geerdete Alufolie packt, wird man nach dem Zusammenbau keine bösen Überraschungen erleben.

Nun hat man die Hauptplatine offen vor sich. Hinten, unterhalb des Lüfters (oder wo er normalerweise sitzt) sieht man einen 40-poligen IC und daneben einen leeren Sockel der gleichen Größe (siehe auch das Foto 1). Der IC ist die CPU (Central Processing Unit). Er muß die Bezeichnung 8088 tragen (z. B. NEC D8088D oder SAB 8088-P). Er wird nun vorsichtig aus seiner Fassung herausgehebelt. Der V20 kann nun mit der Nase nach vorn in die Fassung gedrückt werden. Meist ist es notwendig, die Anschlußbeine etwas zusammenzubiegen. Dies geschieht am einfachsten, indem man den Chip senkrecht zur Tischplatte aufsetzt und die Beinchen nach innen biegt. Hat man den Tausch bewerkstelligt, kann der Zusammenbau erfolgen. Dies geschieht wie die Zerlegung, nur in der umgekehrten Reihenfolge. Auf das korrekte Aufstecken der Kabel sollte besonders geachtet werden. Bei einem folgenden Probelauf dürften sich keine Änderungen zum Verhalten vor dem Austausch ergeben – bis auf die Geschwindigkeit.

Aus dem Takt geraten...

...kann auch der beste Rechner einmal. Im allgemeinen ist das zwar ärgerlich, aber was geschehen ist, ist geschehen. Hier tut sich bei dem SIDECAR jedoch ein Problem auf: Angenommen, der Rechner ist abgestürzt. Man drückt CTRL-ALT-DEL. Nichts geschieht, denn der PC hat die Tastatur „abgehängt". Bei einem normalen Kompatiblen schaltet man jetzt den Rechner kurz aus und der zeigt sich wieder kooperativ. Beim AMIGA-SIDECAR-Gespann drückt man nun CTRL-AMIGA-AMIGA und hat den ganzen Rechner resetet – inclusive AMIGA. Ausschalten würde sogar ein Nachladen der KICKSTART notwendig machen. Daß dies eine zeitaufwendige Prozedur ist, dürfte jedem einleuchten. Doch auch der Reset des AMIGA (beim Booten sendet der AMIGA einen hardwaremäßigen Reset zur Sidecar) hat seine Nachteile: Er dauert auch relativ lange und alle Tasks auf dem AMIGA gehen verloren. Man kann zwar gemütlich alle Daten retten, aber das Nachladen von allen Anwendungen macht beim dritten Mal keinen Spaß mehr.

Nun gibt es beim PC eine Einrichtung, die neben anderen Aufgaben auch für den Reset zuständig ist. Es handelt sich um den Taktgenerator 8284. Hauptaufgabe ist, wie der Name schon sagt, die Takterzeugung. Dabei bietet er dem Rechner zwei verschiedene Taktsignale an: Den Prozessortakt (CLK, Tastverhältnis 1:2) und den Peripherietakt (PCLK, Tastverhältnis 1:1, halbe Frequenz von CLK). Zweite Aufgabe ist die Synchronisation der zum Prozessor gehenden READY-Signale (Pin 3–7,15). Im normalen Betrieb am wenigsten benutzt, für uns in diesem Falle aber sehr nützlich, ist der RESET-Eingang (/RES). Normalerweise wird dieser im SIDECAR nur als Power-Up-Reset genutzt. Beim Einschalten wird durch eine Schaltung etwas verzögert ein Impuls auf /RES gegeben, so daß bei korrekter Spannungsversorgung ein Reset (Kaltstart) gegeben wird. Was diese Powergood-Schaltung macht, kann man aber einfach selbst simulieren. /RES ist ja aktiv 0 („/" bedeutet umgekehrte Logik, d. h. im Ruhezustand beträgt die Spannung auf der Leitung 5 Volt, aktiv ist die Leitung auf Masse gelegt). Man braucht nur einen Taster (Schließer) zwischen /RES (Pin 11) und Masse (GND) zu schalten. Betätigt man jetzt den Taster, wird Pin /RES auf Masse gelegt, also aktiv, und der Taktgenerator gibt einen



prozessorgerechten Reset auf den 8088 (oder V20). Die genaue Pin-Belegung finden Sie in Abbildung 2.

Auf dem Foto 1 sehen sie den IC, um den es sich handelt. Er sitzt wie der Prozessor und der Sockel für den mathematischen Coprozessor unter der Interface-Platine, so daß man das Gerät völlig zerlegen muß, um den 8284 zu erreichen. Wenn man sich freien Zugang zum Taktgenerator verschafft hat, kann man (vorsichtig!!) die beiden Kabel an Pin 9 und Pin 11 anlöten. Die Verwendung eines guten Elektroniklötkolbens (ca. 15 W , geerdete Spitze!) ist sehr zu empfehlen. Die zu benutzenden Anschlüsse kann man anhand von Abbildung 2 finden. Die Abbildung zeigt den IC in der Draufsicht. Beide freien Enden der Kabel werden an einen kleinen Taster gelötet und dieser montiert. Als Platz für den Taster eignet sich zum Beispiel die Abdeckung eines unbenutzten Slots. Eine versehentliche Betätigung ist hier so gut wie ausgeschlossen: Kleine Knöpfe verleiten gute Bekannte und Besucher meist dazu, erst zu drücken und dann zu fragen, was passiert. In dem Slotblech wird er erst gar nicht gesehen. Nach dem Zusammenbau kann man sich dann von der Funktionstüchtigkeit des Tasters überzeugen.

Bridgeboardbesitzer werden sich jetzt fragen: Kann ich auch meinen PC-Teil tunen? 512 kByte RAM hat das Bridgeboard sowieso schon bei Lieferung eingebaut. Da hat Commodore jetzt ein Einsehen gezeigt. Wohl recht spät, denn die zweite Bank RAM-Chips ist noch gesockelt,

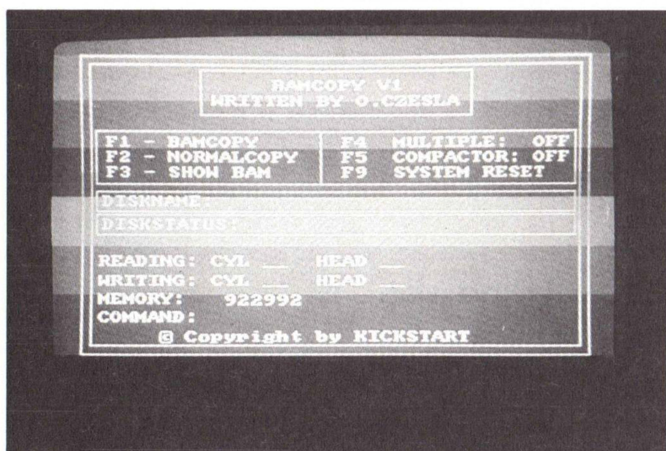
was auf eine Planung mit nur 256 kByte schließen läßt. Der Prozessortausch ist auch hier möglich, gestaltet sich sogar einfacher, da der 8088 sehr leicht zu erreichen ist. Man findet sie recht zentral in der Mitte der Platine gelegen, daneben wieder der Sockel für den Coprozessor 8087. Der Reset-Taster stellt den Bridgeboard-Besitzer vor ernste Probleme. Auf der Platine findet man nämlich keinen 8284. Des Rätsels Lösung liegt in einem speziellen Chip von Commodore. Um den Rechner auf nur eine Platine zu integrieren, mußte man diverse Chips, die in einem normalen Kompatiblen zu finden sind, durch Sonder-ICs ersetzen, die einen wesentlich geringeren Platzbedarf haben als die konventionellen und diejenigen, die mehrere Bausteine zusammenfassen. Da Dokumentationen zu diesen Chips fehlen, war es uns leider bisher nicht möglich, auch für das Bridgeboard eine derartige Lösung anzubieten. Sobald das Problem gelöst ist, werden wir darüber informieren.

(chk)

BAMCOPY 1.0

Anleitung:

Ich habe BAMCOPY V1 für Amigabesitzer geschrieben, die nur das interne Diskettenlaufwerk oder keine Speichererweiterung besitzen.



Nachdem es schon Kopierprogramme gab, die Disketten kopierten und die Daten komprimieren, habe ich mir gedacht, man könne noch mehr Daten einladen, indem man nur das kopiert, was auch auf der Diskette ist.

So dauert das Einladen der Daten einer neu formatierten Diskette mit BAMCOPY-MODUS ca. eine Sekunde.

Ich habe bei dem Kopierprogramm viel Wert auf eine einfache Handhabung und relativ gute Grafik bei möglichst geringem Speicheraufwand gelegt.

Einige Routinen sind in Assembler geschrieben, damit das Kopieren noch schneller geht.

Die unbenutzten Blöcke der Diskette werden beim Schreiben mit Nullen formatiert.

'F1 – BAMCOPY':

Kopiert die Daten, die vom Diskkontroller auch reserviert wurden. (BAM)

Sollte das Kopierprogramm 'DISK NOT VALIDATED' ausgegeben, so war der Diskkontroller aus verschiedenen Gründen nicht in der Lage, alle Blöcke, die benutzt sind, zu reservieren.

In diesem Fall würden Daten verloren gehen, wenn man im BAMCOPY-Modus kopiert, deshalb sollte man die Diskette mit dem NORMALCOPY-Modus kopieren.

'F2 – NORMALCOPY':

Kopiert alle Blocks einer Diskette.

Auch hier kann wahlweise der Kompaktor eingeschaltet werden.

'F3 – SHOW BAM':

Zeigt, welche Blocks einer Diskette belegt sind.

Mit 'SPACE' kann man das Listen solange anhalten bis man die Taste wieder losläßt.

'CAPS LOCK' hält das Listen solange an, bis man wieder CAPS LOCK oder eine beliebige andere Taste drückt.

'ESC' bricht den Listvorgang ab, und dann bringt SPACE einen wieder in den Auswahl-Modus.

'F4 – MULTIPLE COPY':

Wenn sie eine Diskette mehrfach kopieren wollen, so sollten sie den Multiple-Modus einschalten. Dann wird man jedes Mal gefragt, ob man noch einmal schreiben möchte.

'F5 – COMPACTOR':

Man sollte den Kompactor nur einschalten wenn es sich auch lohnt, da das Kopieren sonst länger dauert. Es lohnt sich, wenn man Programme kopiert, die viel Grafik oder Text benutzen.

Anleitung zum Compilieren.

Ich habe ein eigenes makefiles auf die Diskette getan. Zum Compilieren einfach 'Execute makefile' eingeben. Wenn noch Fragen sind, kannst Du anrufen.

Viel Spaß

Oliver Czesla

```
cc NAME.c -e200 +l -s -a -o ram:NAME.asm
as -s4000 -ZAP ram:NAME.asm
ln ram:NAME.o +ccdb -lm32 -lc32
delete ram:NAME.o
copy ram:NAME to df1:examples
```

```
dfl:examples/bam-kick.c

/*************************************************/
/*************************************************/
/***                                          ***/
/***           BAMCOPY VERSION 1.0            ***/
/***                                          ***/
/***    WRITTEN BY OLIVER CZESLA IN 1987      ***/
/***                                          ***/
/***          AMIGA 500/1000/2000             ***/
/***                                          ***/
/***       Copyright KICKSTART 10.87          ***/
/*************************************************/
/*************************************************/

#include <exec/io.h>
#include <exec/ports.h>
#include <libraries/dosextens.h>
#include <graphics/gfx.h>
#include <graphics/rastport.h>
#include <graphics/copper.h>
#include <graphics/view.h>
#include <graphics/gfxbase.h>
#include <graphics/gfxmacros.h>
#include <graphics/text.h>
#include <exec/types.h>
#include <exec/tasks.h>
#include <exec/nodes.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>
#include <devices/trackdisk.h>
#include <exec/lists.h>
#include <exec/memory.h>

#define TD_READ         CMD_READ
#define BLOCKSIZE       TD_SECTOR
#define BSIZE           (TD_SECTOR * NUMSECS)
#define NUMBLOCKS       (NUMCYLS * NUMHEADS * NUMSECS)
#define ROOTBLOCK       (NUMBLOCKS/2)
#define LWORDS          (NUMBLOCKS/32)

extern APTR AllocMem();
extern struct TextFont *OpenDiskFont();
extern struct MsgPort  *CreatePort();
extern struct IORequest *CreateExtIO();

struct Screen        *Screen       = NULL;
struct MsgPort       *diskport;
struct IOExtTD       *diskreq;
struct RastPort      *rp;
struct ViewPort      *vp;
struct UCopList      *cl;
struct IntuitionBase *IntuitionBase = NULL;
struct GfxBase       *GfxBase      = NULL;

struct TextAttr font = { (STRPTR) "topaz.font",8,0,0 };
struct NewScreen MyScreen = { 0,0,320,200,1,0,1,0,CUSTOMSCREEN,&font,0,0,0 };
struct Process *proc, *FindTask();

ULONG  *diskbuffer;

int    i,counter,blocks[1761],dr,cy2,he2,se2,ch1,ch2,ver=0,mul=0;
int    ch1old,ch2old,cylold,heold,seold;
long   wbuf,bfsize;
short  error;
char   buf[60];

UWORD                          /* Farbpalette Vorder- und Hintergrund */
cola[8] = { 0xfbe,0xeae,0xe9e,0xc8d,0xb7d,0xa6c,0x86b,0x75b };
colb[3] = { 0x555,0x444,0x333 };

closescreens()
{
    struct Screen *scr,*scr2;
    struct Window *win,*win2;

    win=IntuitionBase->ActiveWindow;
    while(win!=0)
    {
       win2=win->NextWindow;
       CloseWindow(win);
       win=win2;
    }

    scr=IntuitionBase->FirstScreen;
    while(scr!=0)
    {
       scr2=scr->NextScreen;
       if(scr!=Screen) CloseScreen(scr);
       scr=scr2;
    }
}

box(x,y,x1,y1)
long x,y,x1,y1;
{
  Move(rp,x,y);
  Draw(rp,x1,y);
  Draw(rp,x1,y1);
  Draw(rp,x,y1);
  Draw(rp,x,y);
}

place(x,y,tx)
long x,y;
char *tx;
{
   Move(rp,x,y+6);
   Text(rp,tx,strlen(tx));
}

doit(nr,by)
short nr,by;
{
   diskreq->iotd_Req.io_Length=by;
   if(nr) diskreq->iotd_Req.io_Command=TD_REMOVE; /* Diskinterrupt aus */
   else diskreq->iotd_Req.io_Command=TD_MOTOR;    /* Motor an/aus     */
   DoIO(diskreq);
}

ReadBlock(sector)
long sector;
{
   LONG offset;
   diskreq->iotd_Req.io_Length=BLOCKSIZE;
   diskreq->iotd_Req.io_Data=diskbuffer;
   diskreq->iotd_Req.io_Command=CMD_READ;
   offset=TD_SECTOR*sector;
   diskreq->iotd_Req.io_Offset=offset;
   DoIO(diskreq);
   return (0);
}

rwc(a,cyl, sec, hd, mem)    /* read or write cylinder */
SHORT a,cyl, sec, hd;
LONG mem;
{
   LONG offset;
   short e;
   if(a)
   {
      diskreq->iotd_Req.io_Length=BSIZE;
      diskreq->iotd_Req.io_Command=TD_FORMAT;
   }
   else
   {
      diskreq->iotd_Req.io_Length=BLOCKSIZE;
      diskreq->iotd_Req.io_Command=CMD_READ;
   }
   diskreq->iotd_Req.io_Data=mem;
   offset=TD_SECTOR*(sec+NUMSECS*hd+NUMSECS*NUMHEADS* cyl);
   diskreq->iotd_Req.io_Offset = offset;
   DoIO(diskreq);

#asm
       bchg #1,$bfe001   ; blinken der systemled
#endasm
   e=diskreq->iotd_Req.io_Error;
   if(e!=0) displayerr(e);
}

getbam()
{
   register long check,k,err;
   register short n,j,l,cyl,hel,sel;
   register int v1;
   int bmsect;

   err=OpenDevice(TD_NAME,0,diskreq,0);
   doit(0,1);
   ReadBlock (879);
   ReadBlock (880);  /* rootblock */
   name();
   bmsect = diskbuffer[79];
   check=diskbuffer[78];
   if(check!=-1) place(109,110,"DISK NOT VALIDATED");
   ReadBlock (bmsect);
   doit(0,0);
   counter=v1=0;
   blocks[counter++] = 0;
   blocks[counter++] = 1;
```

```
      for (j=1; j<=LWORDS; j++)
      {
         k = diskbuffer[j];
         for (n=0; n<32; n++)
         if (j<LWORDS || n<30)
         {
            if (~k & 1)
            {
               l = (j-1 << 5) + n + 2;
               cyl=(l/22);
               sel=( l % 22);
               if(sel>=11)
               {
                  sel-=11;
                  hel=1;
               }
               else hel=0;
               v1=(((v1 & 0) | cyl)<<1);
               v1=((v1 | hel)<<4);
               v1=(v1 | sel);
               blocks[counter]=v1;
               counter++;
            }
            k>>=1;
         }
      }
   }

   name()
   {
      long d,adr;
      int i;
      char *c;

      adr= &diskbuffer[108];
      d = (*(int *)adr)>>24;
      if(d>=25) d=25;
      adr+=1;
      for(i=0;i<d;i++)
      {
         c =(char *)(adr+i);
         SetAPen(rp,1);
         Move(rp,93+(i<<3),101);
         Text(rp,c,1);
      }
   }

   showbam()
   {
      int la;
      short cyl,head,sec;

      SetAPen(rp,0);
      RectFill(rp,11,121,312,190);
      getbam();
      for(i=0;i<counter;i++)
      {
         cyl=(blocks[i]>>5);
         head=((blocks[i]>>4)&1);
         sec=(blocks[i] & 15);
         SetAPen(rp,0);
         RectFill(rp,11,121,312,130);
         for(la=0;la<8;la++)
         {
            ScrollRaster(rp,0,1,11,121,312,190);
            wait(1000);
         }
         SetAPen(rp,1);
         sprintf(buf,"CYLINDER: %2d  HEAD: %2d  SECTOR: %2d",cyl,head,sec);
         place(11,180,buf);
         wait(500);
         while(key()==0x3b || key()==0x7f);
         if(key()==0x75) i=counter+2;
      }
      while(key()!=0x7f);
      menus();
   }

   copyme(nr)
   short nr;
   {
      short cb=0,cyl,head,sec,op;
      long adress,mem,old,next,first;
      int v1;

      ch1=ch2=0;
      cy2=he2=se2=ch2=0;
      adress=wbuf;

      give(0);
      if(taste(0x7f)==1) return(1);

      if(nr) getbam(); /* lesen der Bam von der Diskette */
      else             /* simulieren der Bam fuer Cylinder 0-79 */
      {
         counter=0;
         for(cyl=0;cyl<80;cyl++)
            for(head=0;head<2;head++)
               for(sec=0;sec<11;sec++)
               {
                  v1=(((v1 & 0) | cyl)<<1);
                  v1=((v1 | head)<<4);
                  v1=(v1 | sec);
                  blocks[counter++]=v1;
               }
      }

      while(1)
      {
         if(cb) { give(0); if(taste(0x7f)==1) return(1); }

         cb=1;
         memory();
         first=old=AllocMem((long)0x00000408,MEMF_PUBLIC);

         doit(0,1);  /* Motor an */
         while(1)
         {
            cyl=(blocks[ch1]>>5);
            head=((blocks[ch1]>>4)&1);
            sec=(blocks[ch1] & 15);
            sprintf(buf,"%2d  HEAD %2d",cyl,head);
            place(115,134,buf);

            rwc(0,cyl,sec,head,&diskbuffer[0]);
            if(ver) bfsize=comp(&diskbuffer[0],old+8,0x200);
            else
            {
               *(long *) 0x3400=&diskbuffer[0];
               *(long *) 0x3404=(old+8);
               transfer();
               bfsize=0x208;
            }
            *(long *) (old+4) = bfsize;
            bfsize=(((bfsize>>3)<<3)+8);
            FreeMem(old+8+bfsize,0x400-bfsize);
            memory();
            next=AllocMem((long)0x00000408,MEMF_PUBLIC);
            ch1++;
            if(next==0||ch1>=counter)
            {
               *(long *)old=0xccddeeff;
               if(ch1>=counter) FreeMem(next,0x408);
               break;
            }
            *(long *) old = next;
            old=next;
         }

         doit(0,0);
         ch1old=ch1;
         ch2old=ch2;
         cylold=cy2;
         heold=he2;
         seold=se2;

AGAIN:
         ch1=ch1old;
         ch2=ch2old;
         cy2=cylold;
         he2=heold;
         se2=seold;
         give(1);
         if(taste(0x7f)==1) { clearmem(old,first,next); menus(); return(); }

         old = first;
         doit(0,1);
         op=createbuff(old);

         if((ch1>=counter) && cy2<80)
         {
            clear();
            while(1)
            {
               sprintf(buf,"%2d  HEAD %2d",cy2,he2);
               place(115,147,buf);
               rwc(1,cy2,0,he2,adress);
               he2^=1;
               if(he2==0) cy2++;
               if(cy2==80) break;
            }
         }

         doit(0,0);
```

```
        if(mul)
        {
            give(2);
            while(2)
            {
                v1=key();
                if(v1==0x93) break;
                if(v1==0xd5) goto AGAIN;
                if(v1==0x75) { clearmem(old,first,next); menus()
                ; return(); }
            }
        }
        if(ch1>=counter) op=1;
        clearmem(old,first,next);
        if(op)
        {
            place(11,171,"     SUCCESSFUL BACKUP !!    (CTRL)
            ");
            while(key()!=0x39);
            menus();
            return();
        }
    }
}

clearmem(old,first,next)
long old,first,next;
{
        old = first;
        while(1)
        {
            next = *(long *) old;
            bfsize= *(long *)(old+4);
            bfsize=(((bfsize>>3)<<3)+8);
            FreeMem(old,bfsize+8);
            if(next==0xccddeeff) break;
            old=next;
        }
        memory();
}

createbuff(old)
long old;
{
    long next,ad1,adress;
    short cyl,head,sec;
    short u;

    adress=wbuf;
    next=0;
    clear();

    while(1)
    {
        u=1;
        cyl=(blocks[ch2]>>5);
        head=((blocks[ch2]>>4)&1);
        sec=(blocks[ch2] & 15);

        while(u)
        {
            if(cy2==cyl && he2==head && se2==sec)
            {
                ad1=se2*0x200;
                next = *(long *) old;

                if(ver)
                {
                    bfsize= *(long *)(old+4);
                    decomp(old+8,wbuf+ad1,bfsize,0x200);
                }
                else
                {
                    *(long *) 0x3400 = (old+8);
                    *(long *) 0x3404 = (wbuf+ad1);
                    transfer();
                }

                old=next;
                u=0;
                ch2++;
            }
            se2++;
            if(se2==11)
            {
                se2=ad1=0;
                sprintf(buf,"%2d  HEAD %2d",cy2,he2);
                place(115,147,buf);
                rwc(1,cy2,0,he2,adress);
                clear();
                he2++;
                if(he2==2) { he2=0; cy2++; if(cy2==80) return(1)
                ; }
            }
            if(ch2>=ch1 || old==0xccddeeff)
            {
                if(cy2>=80) return(1);
                se2=0;
                while(1)
                {
                    cyl=(blocks[ch2]>>5);
                    head=((blocks[ch2]>>4)&1);
                    sec=(blocks[ch2] & 15);
                    if(sec>=se2 && head==he2 && cyl==cy2) ch2--;
                    else break;
                }
                ch2++;
                ch1=ch2;
                return(0);
            }
        }
    }
}


clear()
{
    *(long *) 0x3400 = wbuf;
    fillbuf();
}

memory()
{
    long memo;
    memo=AvailMem(0);
    sprintf(buf,"%7d",memo);
    place(83,160,buf);
}

menus()
{
    SetAPen(rp,0);
    RectFill(rp,0,0,319,200);

    SetAPen(rp,1);
    box(0,0,319,199);
    box(6,6,313,193);
    box(77,10,243,41);
    box(9,52,150,87);
    box(150,52,310,87);
    box(9,92,310,105);
    box(9,107,310,120);
    place(121,17,"BAMCOPY V1");
    place(84,28,"WRITTEN BY O.CZESLA");
    place(16,56,"F1 - BAMCOPY");
    place(16,66,"F2 - NORMALCOPY");
    place(16,76,"F3 - SHOW BAM");
    place(163,56,"F4  MULTIPLE:");
    place(163,66,"F5  COMPACTOR:");
    place(163,76,"F9  SYSTEM RESET");
    place(13,95,"DISKNAME:");
    place(13,110,"DISKSTATUS:");
    place(11,134,"READING: CYL __  HEAD __");
    place(11,147,"WRITING: CYL __  HEAD __");
    place(11,159,"MEMORY:");
    place(11,171,"COMMAND:");
    place(11,184,"       c Copyright by KICKSTART");
    memory();
    state();
}

main()
{
    short cyl,head,sec,v,m;

    openstuff();
    closescreens();
    proc=(struct Process *)FindTask("File System");
    proc->pr_WindowPtr=-1;
    menus();

    while(1)
    {
        v=key();
        if(v==0x5b) showbam();
        if(v==0x5f) copyme(1);
        if(v==0x5d) copyme(0);
        if(v==0x59)             /* multiple */
        {
            mul^=1;
            state();
            wait(50000);
        }

        if(v==0x57)             /* compact */
        {
            ver^=1;
            state();
            wait(50000);
        }
```

```
        if(v==0x4f) reset();
    }
}

#asm
          public _key, _fillbuf, _transfer, _reset
_key  moveq  #0,d0
      move.b $bfec01,d0
      rts

_fillbuf
      move.w #$57f,d0
      move.l $3400,a0
l1    clr.l  (a0)+
      dbf    d0,l1
      rts

_transfer
      move.w #127,d0
      move.l $3400,a0
      move.l $3404,a1
l2    move.l (a0)+,(a1)+
      dbf    d0,l2
      rts
_reset
      move.l  #$fc0000,$20 ; Reset
      rts
#endasm

give(nr)
short nr;
{
   if(nr==0) place(83,171,"INSERT SOURCE              ");
   else if(nr==1) place(83,171,"INSERT DESTINATION         ");
        else place(83,171,"WRITE BUFFER AGAIN (Y/N)");
}

wait(olcz)
long olcz;
{
   long czol;
   for(czol=0;czol<olcz;czol++);
}

openstuff()
{
   int a,b,c,d,i;

   if(!(IntuitionBase=(struct IntuitionBase *)OpenLibrary("i
   ntuition.library",0))) exit(1);
   if(!(GfxBase=(struct GfxBase *)OpenLibrary("graphics.libr
   ary",0))) exit(2);
   if(!(Screen = (struct Screen *)OpenScreen(&MyScreen))) ex
   it(3);
   diskbuffer=AllocMem (BLOCKSIZE, MEMF_CLEAR | MEMF_CHIP);
   wbuf=AllocMem (BSIZE,MEMF_CLEAR|MEMF_CHIP);
   if(!(cl=AllocMem(sizeof(struct UCopList),MEMF_PUBLIC|MEMF
   _CHIP|MEMF_CLEAR))) exit();

   rp= &Screen->RastPort;
   vp= &Screen->ViewPort;

   CWAIT(cl,0,0);              /* create the copperlist */
   CMOVE(cl,(*((UWORD *)0xdff140)),0); /* no pointer */
   CMOVE(cl,(*((UWORD *)0xdff142)),0);
   d=b=a=c=0;

   for(i=16;i<200;i+=16)                /* colors */
   {
      CMOVE(cl,(*((UWORD *)0xdff182)),cola[b]);
      if(d==0)
      {
         CMOVE(cl,(*((UWORD *)0xdff180)),colb[a]);
         a++;
         if(a==3) a=0;
      }
      CWAIT(cl,i,0);
      c++;
      d++;
      if(d==2) d=0;
      if(c<7) b++;
      if(c>=7 && c<=16) b--;
   }

   CWAIT(cl,200,0);
   CMOVE(cl,(*((UWORD *)0xdff180)),0x0000);
   CMOVE(cl,(*((UWORD *)0xdff182)),0x0000);
   CEND(cl);
   vp->UCopIns=cl;
   RethinkDisplay();
   SetDrMd(rp,JAM2);
   diskport=CreatePort(0,0);
   diskreq=(struct IOExtTD *)CreateExtIO(diskport,sizeof(str
   uct IOExtTD));
   error = OpenDevice(TD_NAME,0,diskreq,0);
   doit(1,0);   /*  diskinterrupt  s p e r r e n  */
}

state()
{
   if(mul)  place(283,56,"ON ");
   else place(283,56,"OFF");
   if(ver)  place(283,66,"ON ");
   else place(283,66,"OFF");
}

displayerr(e)
short e;
{
   sprintf(buf,"DISKERROR: #%02ld      ",e);
   place(109,110,buf);
}

taste(ch)
int ch;
{
   int o;
   while(1)
   {
      o=key();
      if(o==ch)
      {
         place(83,171,">>> PLEASE WAIT <<<      ");
         return(0);
      }
      if(o==0x75) { menus(); return(1); }
   }
}



/****************************************************/
/* ELECTRONIC ARTS COMPACT AND DECOMPACT ROUTINE */
/*        BY STEVE SHAW AND JERRY MORRISON        */
/****************************************************/

#define GetByte()         (*source++)
#define PutByte(c)        { *dest++=(c); ++putSize; }
#define OutDump(nn)       dest = PutDump(dest,nn)
#define OutRun(nn,cc)     dest = PutRun(dest,nn,cc)
#define MinRun            3
#define MaxRun            128
#define MaxDat            128
#define DUMP              0
#define RUN               1

LONG putSize;
char cbuf[256];

BYTE *PutDump(dest, nn)
register BYTE *dest;
register long nn;
{
   register long i;
   PutByte(nn-1);
   for(i=0;i<nn;i++)
   PutByte(cbuf[i]);
   return(dest);
}

BYTE *PutRun(dest,nn,cc)
register BYTE *dest;
register long nn,cc;
{
   PutByte(-(nn-1));
   PutByte(cc);
   return(dest);
}

comp(pSource,pDest,rowSize)
BYTE *pSource,*pDest;
LONG rowSize;
{
register     BYTE  *source,*dest;
register     char  c,lastc='\0';
BOOL         mode=DUMP;
register     short nbuf=0;
register     short rstart=0;

   source=pSource;
   dest=pDest;
   putSize=0;
   cbuf[0]=lastc=c=GetByte();
   nbuf=1;
   rowSize--;
```

```
	for (;rowSize;--rowSize)
	{
	   cbuf[nbuf++]=c=GetByte();
	   switch (mode)
	   {
	      case DUMP:
	                  if(nbuf>MaxDat)
	                  {
	                     OutDump(nbuf-1);
	                     cbuf[0]=c;
	                     nbuf=1;
	                     rstart=0;
	                     break;
	                  }
	                  if(c==lastc)
	                  {
	                     if (nbuf-rstart>=MinRun)
	                     {
	                        if (rstart>0) OutDump(rstart);
	                        mode = RUN;
	                     }
	                     else if(rstart==0) mode=RUN;
	                  }
	                  else rstart=nbuf-1;
	                  break;
	      case RUN:
	                  if((c!=lastc)||(nbuf-rstart>MaxRun))
	                  {
	                     OutRun(nbuf-1-rstart,lastc);
	                     cbuf[0]=c;
	                     nbuf=1;
	                     rstart=0;
	                     mode=DUMP;
	                  }
	                  break;
	   }
	   lastc=c;
	}  /* ende der for schleife */

	switch(mode)
	{
	   case DUMP: OutDump(nbuf); break;
	   case RUN:  OutRun(nbuf-rstart,lastc); break;
	}
	return(putSize);
}


#define UGetByte()      (*source++)
#define UPutByte(c)     (*dest++ = (c))

decomp(pSource, pDest, srcBytes0, dstBytes0)
BYTE *pSource, *pDest;
WORD srcBytes0, dstBytes0;
{
	register BYTE *source = pSource;
	register BYTE *dest   = pDest;
	register WORD n;
	register BYTE c;
	register WORD srcBytes = srcBytes0,
	              dstBytes = dstBytes0;
	BOOL error = TRUE;

	while(dstBytes>0)
	{
	   if((srcBytes-=1)<0)  return(error);
	   n=UGetByte();
	   if(n>=0)
	   {
	      n+=1;
	      if(((srcBytes-=n)<0)||((dstBytes-=n)<0)) return(err
or);
	      do {  UPutByte(UGetByte()); } while (--n>0);
	   }

	   else if(n!=-128)
	   {
	      n=-n+1;
	      if(((srcBytes-=1)<0)||((dstBytes-=n)<0)) return(err
or);
	      c = UGetByte();
	      do {  UPutByte(c);  } while (--n>0);
	   }
	}
	error=FALSE;
	return(error);
}
```

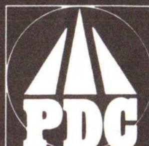
AMIGA WARE

Deluxe Video 1.2	265,00
Lattice C Compiler	415,00
Barbarian	80,00
Karate King	49,90
Rocket Attack	29,90
Shooting Star	29,90
Space Fight	29,90
Galileo	135,00
Forms in Flight	185,00
Sky Fighter	59,90
Fontset 1	70,00
Video Vegas	90,00
Digipaint	140,00
The Calligrapher	195,00
Analytic Art	110,00
Videoscape 30	385,00
Sculpt 3D	195,00
Aegis Draw Plus	485,00
Aegis Animat. m. Images	265,00
Aegis Sonix	160,00
Alien Fires	95,00
Aztec C (Developers)	620,00
Bards Tale	95,00
Bureaucracy	90,00
Butcher	75,00
Challenger	29,90
Cruncher Factory	29,90
dbMan	295,00
Deep Space	80,00
Defender of the Crown	95,00
Deja Vu	95,00
Deluxe Art Parts Vol II	70,00
Deluxe Paint II	265,00
Deluxe Video	220,00
Digiview	410,00
Feary Tale Adventure	90,00
Flight Simulator II	95,00
Flight Sim. Scenery Disc (East USA)	55,00
Guild of the Thieves	85,00
Hollywood Poker	49,90
Kings Quest I	95,00
Kings Quest II	95,00
Kings Quest III	95,00
Marauder II	80,00
MIDI Interface (ECE)	140,00
MIDI Interface (Mimetics)	115,00
Modula 2 (Regular Version)	195,00
Modula 2 (Develop. Vers.)	325,00
Modula 2 (Commercial Vers.)	630,00
Mouse Pad	14,50
Prism	135,00
Textomat Amiga	99,00
Pro MIDI Studio	345,00
Publisher 1000	390,00
Roadwar 2000	95,00
Sound Sampler	225,00
Starglider	75,00
Starfleet 1	120,00
The Surgeon	149,00
TV Text	225,00

UND WEITERE PROGRAMME

Alle Preise in DM

Porto bei Vorkasse 4,– DM
bei Nachnahme 6,– DM



Auslandsbestellungen nur gegen Vorkasse + 10,– DM Porto

Bestellservice 9-18.30
06172/24748 + 20799

PDC GmbH · Louisenstr. 115 · Passage Alter Bahnhof · 6380 Bad Homburg · Tel. 06172/24748

Tips & Tricks

TIP 1:

Viele Amiga-500- und 2000-Besitzer werden sich schon über den fehlenden Apostroph, den man ja in Basic für das Einfügen von Kommentaren und im CLI als Pattern braucht, gewundert haben. Man kann zwar die Taste rechts neben dem „?" benutzen und danach die Space-Bar drücken. Nun erscheint ein Apostroph auf dem Bildschirm, welchen z. B. der Basic-Interpreter nach dem Drücken von Return anerkennt, aber nach Starten des Programms in diesem Apostroph einen Syntax-Error „erkennt".

Abhilfe schafft hier das Drücken der Tasten „Alt" und „Ä" gleichzeitig. Man sollte sich auch gleich die Taste „Ü" und die Taste „★" merken, welche in Verbindung mit der „Alt"-Taste ein „[" oder „]" ergibt. Diese beiden Klammern werden im Amiga-Basic öfter als Steuerzeichen für die Escape-Sequenzen des Druckers gebraucht.

TIP 2:

Die meisten meiner Bekannten, die einen Amiga 2000 kauften, kamen mit ihrer Hardwareuhr nicht zurecht. Und zwar zeigte die Uhr nach erneutem Ein/Ausschalten immer die falsche Uhrzeit und das falsche Datum an. Im Benutzerhandbuch ist die Handhabung der Hardwareuhr auch nicht erklärt. Hier der CLI-Befehl, um die Uhr dauerhaft einzustellen: Die Uhr muß natürlich gestellt sein. Nun folgenden Befehl im CLI-Window eingeben: SETCLOCK OPT SAVE. In der Startup-Sequence der Workbench müßte eigentlich der Befehl SETCLOCK OPT LOAD enthalten sein. Wenn nicht, sollte man dies nachholen.

Es genügt also, SETCLOCK OPT SAVE einmal einzugeben. SETCLOCK OPT LOAD muß aber nach jedem erneuten Booten an den Rechner geschickt werden und gehört deswegen am besten in die Startup-Sequence.

TIP 3:

Es gibt ja Programme, die auf einem Amiga 2000 nicht lauffähig sind, da dieser über 1 Megabyte Speicherplatz verfügt. Nun kann man das Programm NoFastMem laden, um 512 kB „abzuklemmen". Dann kann man auch diese Programme laden, allerdings nur diejenigen, die über eine Programm-Icon verfügen und nicht gebootet werden müssen. Da aber die meisten Spiele Boots sind, geht das nicht. Dort muß man die Startup-Sequence ändern bzw. ergänzen.

Als Beispiel hier die Anleitung zum Ändern der Startup-Sequence von Flip Flop: Im CLI-Window „ED S/STARTUP-SEQUENCE" eingeben. Nun wird der Editor „Ed" mit der Startup-Sequence geladen. Jetzt gleich in der obersten Zeile RETURN drücken, um ganz oben eine leere Zeile für die Eingabe freizubekommen. Dann eingeben: RUN NOFASTMEM. Nun die „Esc"-Taste und dann „x" drücken. Dann wird die neue Startup-Sequence abgespeichert. Nun versuchen Sie, Flip Flop neu zu booten. Als erstes wird nun NoFastMem geladen und dann Flip Flop. Dies funktioniert bei fast allen Programmen, bei denen das Directory bzw. die Startup-Sequence nicht aus Kopierschutzgründen „versteckt" wurden.

Bei manchen Programmen ist allerdings noch der WAIT-Befehl notwendig. Bei Flip Flop ist dies nicht notwendig, da nach dem Befehl „RUN NOFASTMEM" noch ein paar „ECHO-Befehle" kommen, die dem Computer Zeit lassen, das NoFastMem ganz zu laden, bevor der Befehl Flip Flop kommt, der ja das Spiel lädt.

Bei den Programmen, bei denen kein „ECHO-Befehl" nach dem RUN NOFASTMEM kommt, sondern sofort das Hauptprogramm geladen wird, drückt man nach der Eingabe von RUN NOFASTMEM die Return-Taste und gibt in der nächsten Zeile dann noch den CLI-Befehl „WATT 10" ein. Dieser veranlaßt den Computer, dem Programm NoFastMem 10 Sekunden Zeit zum Laden zu geben. Danach wieder „Esc" und „x" zum Abspeichern drücken.

Sollte auf Ihrer Diskette der „RUN-" oder „WAIT"-Befehl nicht vorhanden sein, müssen Sie von Ihrer Workbenchdiskette aus der C-Schublade heraus auf Ihre Programmdiskette kopieren.

Auf diese Art habe ich bis jetzt jedes meiner Programme zum „Laufen" gebracht.

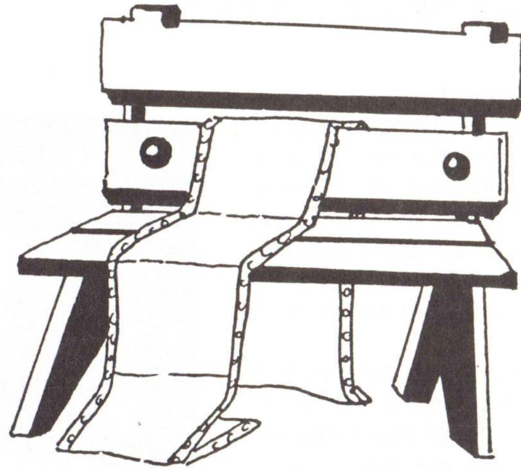
TIP 4:

Sollten Sie mal schnell etwas eintippen wollen, das sofort auf dem Drucker ausgegeben werden soll, und nicht einen Editor bzw. eine Textverarbeitung laden wollen, geben Sie folgenden Befehl ein:
COPY ★ to PRT:

Das „★" ist das Symbol für das aktuelle CLI-Window. Nach dieser Eingabe verschwindet der CLI-Prompt „1)". Man sieht nun nur noch den Cursor auf dem Bildschirm. Nun wird alles, was Sie eingeben, durch Druck auf die Return-Taste ausgedruckt. Durch Drücken der Tastenkombination „Ctrl – /" kommen Sie in den „normalen" CLI-Modus zurück.

DATAMAT

Eine Datenbank für Einsteiger



Die einfache Bedienbarkeit und vor allem der günstige Preis machen diese Datenbank interessant für den Anwender. Der zur Datenverwaltung notwendige Maskenaufbau ist schnell durchgeführt und die Eingabe der eigenen Datei kann beginnen. Für die am häufigsten benutzten Befehle gibt es entsprechende Piktogramme, die leicht erkennbar die Aktion darstellen. Unser Test zeigt die Mögichkeiten und Grenzen dieser Datenbank auf.

Der erste Schritt zum Aufbau einer eingenen Datenbank ist das Erstellen der Eingabemaske. Dieser Vorgang ist dank der einfachen Bedinung von DATAMAT schnell erledigt, allerdings sollte man sich vorher überlegen, welche Punkte in der Datei vorkommen sollen, und auch ein gewisse Ordung sollte vorgenommen werden. Es ist zwar jederzeit möglich, diesen Aufbau – selbst grundlegend – zu ändern, allerdings ist dies, trotz des komfortabeln Maskeneditors, mit einiger Arbeit verbunden.

Die einzelnen Datenfelder sind in der Voreinstellung für eine Länge von 35 Zeichen vorgesehen, aber auch das ist später schnell umgestellt. Recht wichtig ist bei jedem Feld der jeweilige Typ, wobei Text-, Datum-, Zeit-, Zahlen-, IFF-Format-(für IFF-Bilder) und Auswahlfelder unterschieden werden.

Nachdem alle Datenfelder und deren Typen eingegeben sind, ist die Maskenerstellung prinzipiell vollzogen. Änderungen und Verfeinerungen können später noch angebracht werden. Nun erfolgt eine unter Umständen wichtige Prozedur: die Passworteingabe. Wenn Ihre wertvolle Datei vor der Veränderung oder gar dem Einblick fremder Wesen geschützt werden soll, dann haben Sie nun die einmalige Chance, dafür ein Passwort vorzusehen. Dabei werden zwei Schutzstufen unterschieden. Das Benutzerwort berechtigt dabei nur zum Arbeiten mit der Datei, nicht zu Manipulationen an den Masken und den Datensätzen. Dies ist nur dem Operator vorbehalten. Der Benutzer kann nur in den Daten herumblättern.

Die Suche nach Daten...

Das Suchen nach einem bestimmten Datensatz ist auch in einer 'elektronischen' Datenbank eine mühsame Sache, wenn man dazu jeden Satz anschauen müßte und die Daten einen gewissen Umfang erreicht haben. Die vielfältigen Suchfunktionen sind hierbei eine große Hilfe. In jedes Feld der Maske kann dazu ein Suchbegriff eingetragen werden. Wenn man den genauen Begriff nicht weiß, dann helfen Jokerzeichen oder die Eingabe mehrerer Alternativen. Die Suche kann sich sogar über den ganzen Datensatz ausweiten, wenn das betreffende Feld nicht bestimmbar ist.

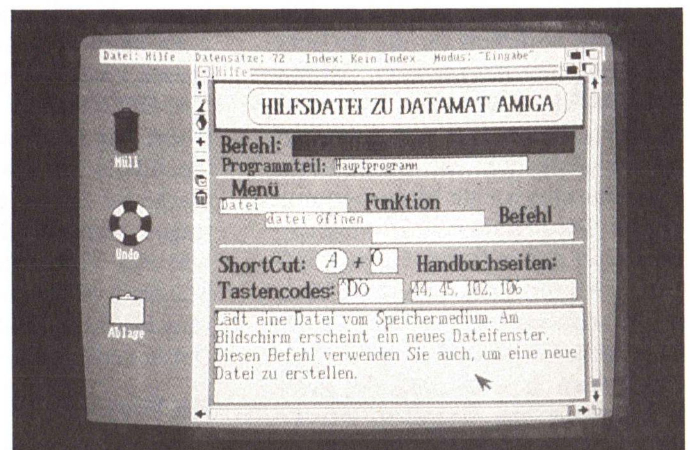
Sortieren mit Indexfeldern

Es ist fast immer sinnvoll, eine Datei, die schon über eine längere Zeit bearbeitet und verändert wurde, zu sortieren. Natürlich kann eine Datei nicht nach allen Feldern gleichzeitig geordnet sein. Deshalb wird ein Feld aus Index festgelegt. Nach diesem Feld wird nun der Datensatz sortiert. Besonders für die Ausgabe der Daten ist eine solche Ordnung wünschenswert, um auch dann noch bestimmte Begriffe wiederfinden zu können. Für jede Datei können bis zu 80 verschiedene Indexfelder bestimmt werden, so daß es möglich ist, eine Datei nach allen ihren Feldern zu sortieren.

Allerdings ist es nicht möglich, eine Mehrfachgliederung der Daten vorzunehmen: Wenn ein Feld nach einem Kriterium sortiert ist und nun z. B. mehrerer hundert Personen im PLZ-Gebiet 6000 wohnen, dann kann diese Personengruppe nicht noch nach dem Namen geordnet werden.

Selektieren von Datensätzen

Wenn aus einer bestehenden Datei eine gewisse Datengruppe herausgesucht werden soll, um sie speziell zu bearbeiten (z. B. ausdrucken), dann wird diese Menge selektiert. Dies



Aufbau einer Datei

kann dann ein Zahlenbereich (PLZ von 6000 – 6230) oder ein Buchstabenbereich (Namen von A – L) sein. Fast alle Funktionen der Datenbank beziehen sich nun auf die selektierten Datensätze, so daß diese weiter bearbeitet und ausgegeben werden können.

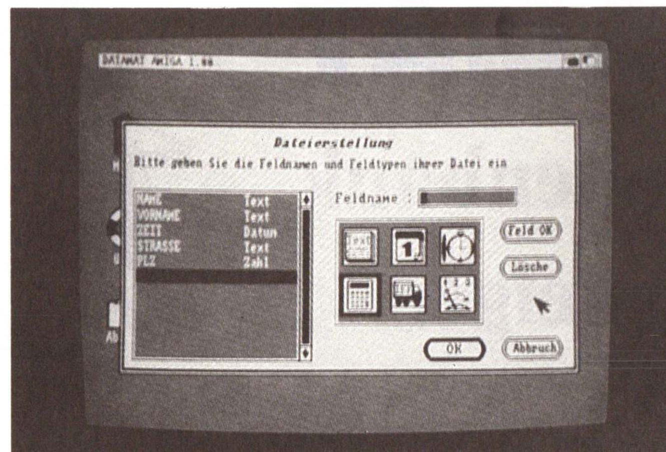
Besonders wichtig ist diese Option für die Erstellung von Serienbriefen, bei denen aus einer großen Datei nur ein bestimmter Personenkreis herausgesucht wird. Aus dieser Datei können dann noch bestimmte Felder und deren Reihenfolge ausgewählt werden, falls nur sie für den Brief relevant sind. Diese Auswahl wird dann als Datei auf Diskette abgelegt und kann von einem Textverarbeitungsprogramm für die Serienbriefschreibung verwendet werden. Kürzere Schriftstücke können auch direkt mit DATAMAT erstellt werden.

Was verbirgt sich hinter den Masken?

Die erste Eingabedatei (siehe Bild 1) ist ebenfalls in einer Maske eingegeben worden. Nun ist es so, daß zu jedem Programm mehrere Masken erstellt werden können, die jeweils verschiedene Teildaten der Datei darstellen können. Es ist aber auch denkbar, daß mehrere Dateien die gleiche Ein- oder Ausgabemaske verwenden. Doch wenden wir uns den Maskeneditoren zu, denn es gibt beim DATAMAT gleich drei davon: den Bildschirm-, Listen- und Druckermaskeneditor.

Besonders leistungsfähig und komfortabel ist dabei der Bildschirmmaskeneditor. Hier wird vor allem die Position und die Länge der Eingabefelder bestimmt, aber auch Schriften und grafische Elemente können hier in die Datenbank eingebracht werden. Besonders die grafischen Elemente sind natürlich für die optische Gestaltung sehr gut geeignet. Befehle wie Linie, Kreis, Rechteck, Farbwahl, Linientyp, Füllmuster und Schriftart sind nur einige aus dem großen Angebot, das eher an ein Malprogramm erinnert. Die gestalterischen Möglichkeiten sind dadurch fast unerschöpflich, wie man auch der Gestaltung der Hilfsdateien von DATAMAT ansehen kann (siehe Bild 3).

Sehr wichtig sind auch die Feldattribute, die hier festgelegt werden können. Normalerweise können in



Gestaltungsmöglichkeiten für eine Datei

ein Datenfeld relativ freie Eingaben gemacht werden. Doch öfter ist es sinnvoll, bestimmte Überprüfungen für ein Feld vorzusehen. Dazu werden die Typen Überprüfungs-, Automatik-, Repetier- und Pflichtfeld verwendet.

Der Listen- und der Druckermaskeneditor sind nicht so umfangreich ausgestattet, aber auch sie erlauben einige grafische Eingriffe und z. B. den Ausdruck auf Adressetiketten oder Formulare.

Auf jeden Fall können dieser Dateiverwaltung sehr gute Maskeneditorfunktionen bestätigt werden, die viele Möglichkeiten bieten und dabei vielen Anforderungen gerecht werden.

Was kommt dabei raus...

Als erstes lassen sich die Parameter für die Ein-/Ausgabe festlegen. Dazu gehören neben dem Laufwerk und dem Datenordner auch die Codes für den Feld- und Satztrenner, so daß man sich den verschiedenen Serienbriefprogrammen anpassen kann. Die Ausgabe kann auf Drucker, Bildschirm und Floppy erfolgen. Auch Daten anderer Dateiverwaltungen können eingelesen werden.

Es kann aber auch der Datenaustausch zwischen verschiedenen DATAMAT-Dateien vollzogen werden. Dabei werden bestimmte Felder der Quelldatei ausgewählt, die dann in markierte Felder der Zieldatei übertragen werden. Diese Option trägt sehr zur Flexibilität der Datenbank bei, denn es kommt öfters vor, daß in verschiedenen Dateien die gleichen Daten erwünscht sind.

Ein Wort zum Komfort

DATAMAT ist, wie aus den Bericht bereits hervorgeht, eine sehr bedienerfreundliche Dateiverwaltung. Alle Funktionen werden mit der Maus und über Menüs vorgenommen. Zusätzlich gibt es die Möglichkeit, alle wichtigen Funktionen über Tastenkombinationen, sogenannte Shortcuts, abzusetzen; sogar eine Funktionstastenprogrammierung ist durchführbar. Auch an vielen anderen Stellen zeigt sich, daß hier großes Augenmerk auf die leichte und komfortable Bedienung gelegt wurde.

Deshalb...

Hohe Leistungsfähigkeit, leichte Bedienung und viel Komfort zeichnen diese Dateiverwaltung aus. Dazu kommt noch eine große Anzahl von Spezialfunktionen, die sicher nur von den wenigsten Anwendern ausgeschöpft werden können. Zu einem Preis von DM 99,– ist das eine gute Empfehlung, die sich auch oder gerade für den 'kleinen Geldbeutel' und Einsteiger eignet.

(mn)

+ bis zu 8 Dateien gleichzeitig
+ Suchen/Sortieren/Selektieren
+ umfangreiche Maskeneditoren
+ Passwortabfrage
+ deutsche Bezeichnungen/Anleitung
+ kein Kopierschutz
+ günstiger Preis

- teilweise sehr verspielte Gestaltung des Programms
- keine Mehrfachgliederung

Anbieter: DATA BECKER

Preis: DM 99,–

EMERALD MINE

Die unterirdischen Höhlen der „Emerald Mine" erscheinen vertraut unvertraut. Ohne Brille könnte der kurzsichtige Amiga-Freak dieses Spiel mit „Boulder Dash" verwechseln – nur ist es besser.

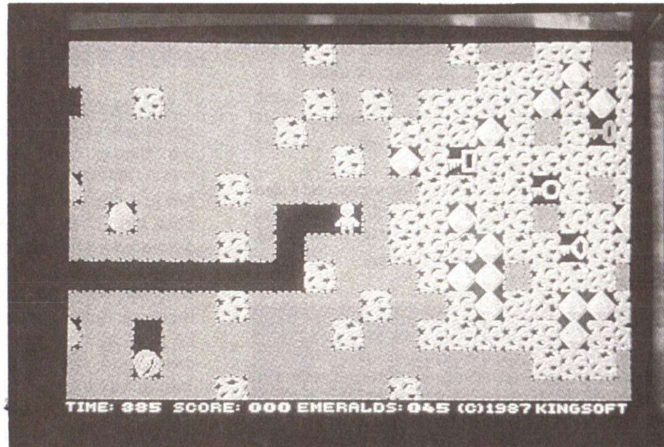
Tatsächlich ist die dem Spiel zugrundeliegende Idee mit ihrem zarten Alter von zwei Jahren für Computerfreunde schon eine angegraute Dame. Diesen Nachteil macht sie allerdings durch einige bemerkenswerte Schönheitsoperatinen wett.

Im Gegensatz zu bisherigen Spielen dieser Art gibt es einen Zwei-Spieler-Modus, der erreichte Hi-Score kann abgespeichert werden und man kann jedes Level, das man einmal geschafft hat, überspringen.

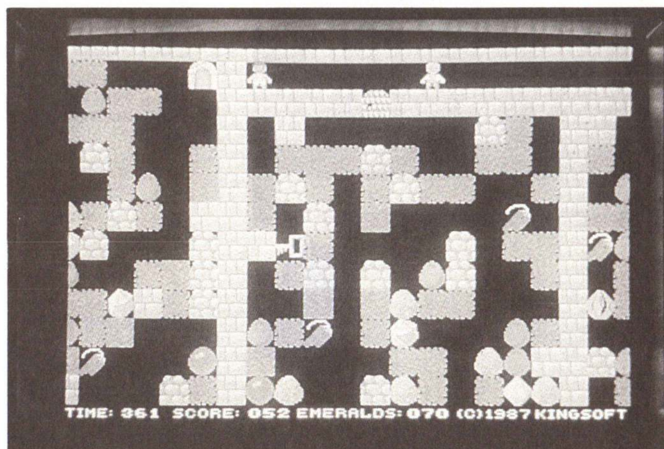
Die Aufgabe des Spielers ist es, aus einem unterirdischen Labyrinth zu entkommen, in dem es von Monstern nur so wimmelt. Herabstürzende Steine und explodierende Bomben haben nicht nur die Funktion von Hindernissen. Sie werden sehr oft auch als Hilfsmittel benötigt, um an Diamanten heranzukommen. Erst wenn der Spieler eine Anzahl von Diamanten gesammelt hat, öffnet sich die magische Tür zum nächsten Level. Grafik und Scrolling dieses Spiels sind überzeugend. Titelmelodie und Sound sind digitalisiert.

Das Spiel funktioniert nur mit Joystick. Auch die Menüs lassen sich nicht per Maus steuern. Insgesamt hinterließ Emerald Mine einen guten Eindruck. Es ist eine gute Alternative zu den schon zahlreich vorhandenen Ballerspielen.

(Peter Schulz / cpl)



Auf der Suche nach den Diamanten muß so manches Hindernis umgangen werden



Höhere Levels benötigen eine gute Lösungsstrategie, sonst gibt es keine Rettung

IDEE:

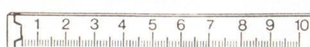


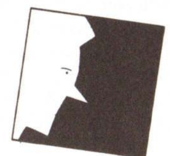
SOUND:



GRAFIK:





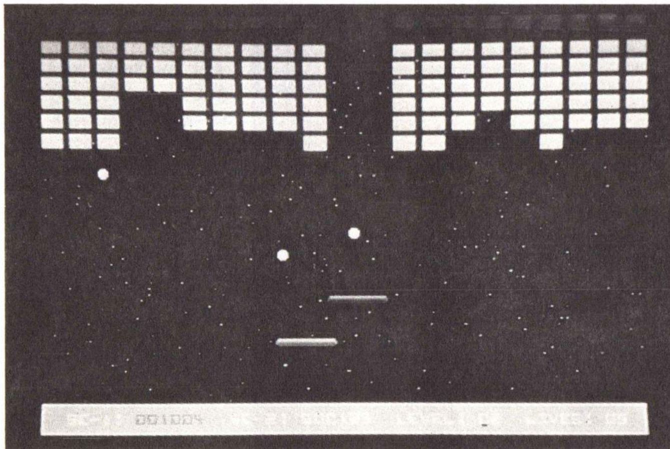


Spiele

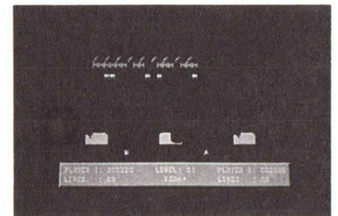
MINDBREAKER

Der Spielaufbau ist mit wenigen Worten beschrieben: Eine Mauer, ein bewegbarer Schläger und dazu noch eine Kugel. Richtig – es handelt sich um eine Version des Spielhallenklassikers 'Breakout'. Doch es gibt einige Neuerungen: In bestimmten Abständen kommen neue Kugeln hinzu, so daß man, mit etwas Geschick, mit mehreren Kugeln die Mauer entfernen kann. Das Spiel sollte man nur spielenm, wenn man genügend Zeit hat oder sehr geschickt mit dem Schläger umgehen kann, denn es läßt keine Ruhe, bis man es endlich geschafft und auch den letzten Stein der Mauer entfernt hat. Danach winkt endlich ein neuer Level – und das Spiel geht weiter.

Hier zeigt sich, daß spannende und gute Spiele nicht unbedingt teuer sein müssen. MINDBREAKER verdient deshalb eine gute Note.



SPACE FIGHT



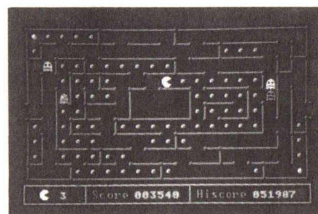
Das Spiel SPACE FIGHT läuft schon seit vielen Jahren unter anderen Namen (zum Beispiel SPACE INVADERS) in den Spielhallen und auf fast allen Homecomputern. Man führt ein Geschütz, das sich nur auf der unteren Linie bewegen läßt. Mehrere Reihen von fremden Wesen kommen nun von oben nach unten. Von Reihe zu Reihe werden sie schneller. Es darf aber keiner unten ankommen, sonst hat man verloren. Vor Jahren konnte man sich noch an einem solchen Spiel erfreuen, aber die heutigen Computer-Benutzer sind einen anderen, weitaus besseren Spielestandard gewohnt. Doch obwohl die Grafik mehr als dürftig ist, geht von der Spielidee eine gewisse Faszination aus, die sich besonders im Zwei-Spieler-Modus entwickelt.

MINDBREAKER, PAC-BOY, SPACE FIGHT
Preis: je 29.90 DM
Bezugsquelle: PDC, Bad Homburg

PAC-BOY

PAC-BOY ist nichts anderes als das berühmte Spiel PAC-MAN. Man befindet sich in einer Art Irrgarten, in dem jeder Weg mit Punkten gepflastert ist, die man auffressen muß. Dabei muß man auf Geister achten, die sich ebenfalls im Irrgarten befinden. Die Geister kann man allerdings auch fressen, wenn man einen der bestimmten Punkte gefressen hat, die sich in den Ecken befinden. Sind alle Punkte beseitigt, kommt man in den nächsten Level. Außer, daß hier die Geister etwas schneller werden, bleibt alles beim alten.

Die digitalisierte Musik am Anfang trügt ein wenig, denn was dann folgt, ist eine mäßige Grafik – und ein Spiel, das sowieso schon jeder kennt. Im Vergleich zu anderen PAC-MANs ist PAC-BOY nicht so gut gelungen, schon gar nicht für den Amiga mit seinen besonderen Grafikfähigkeiten!

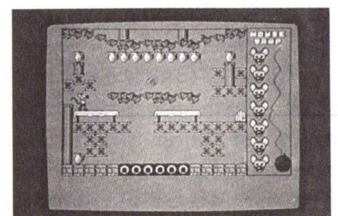


Alles Käse

Irgend jemand hat wieder einmal den Käse zum Bahnhof gerollt. Und da Tränen zwar Steine und Männer erweichen, aber keinen echten Emmentaler aufhalten können, heißt das Motto: Selbst ist die Maus – und nichts wie hinterher. Was sich lapidar unter der Kategorie „Hüpf- und Springspiel" abheften ließe, heißt hier „Mousetrap" und verlangt Geschick im Umgang mit dem Joystick und ein exaktes Timing. Die Jagd nach dem gelben Leckerbissen führt durch eine Unzahl von Räumen. Sie sind voll von unfreundlichen Zeitgenossen und mäusefeindlichen Hindernissen. Unsere Maus hat zwar nur zwei Leben weniger als die bundesdeutsche Durchschnittskatze, aber die Mausefallen fordern sehr schnell ihren Tribut.

Die Spielidee hebt „Mousetrap" angenehm von der „Schlag drauf und Schluss"-Welle ab, die sich im Moment breit macht. Graphik und Sound sind guter Durchschnitt. „Mousetrap" ist ein gutes Beispiel dafür, das sich das Prädikat „Billigspiel" nicht auf die Qualität, sondern auf den Preis bezieht. Wer gerne etwas springen läßt, handelt hier ganz sicher nicht mit Zitronen.

cpl



Gib all deine Hoffnung auf bevor du anfängst:

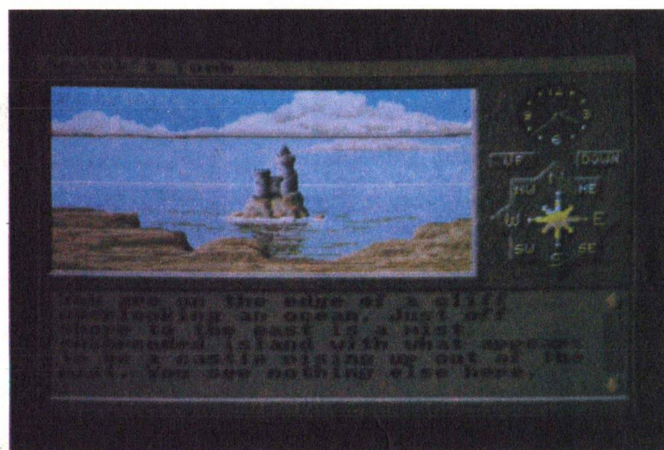
ARAZOK'S TOMB

Als Sensationsreporter des 'Herald Tribune' erhalten Sie die Nachricht, daß Onkel Jason von seiner letzten Ausgrabung nicht mehr zurückgekehrt sei. Sofort machen Sie sich auf den Weg nach Schottland. Das Ziel heißt 'CAER ARAZOK', und es wird erzählt, daß an diesem Ort schon seltsame Dinge passiert sind. Vor langer Zeit soll dort die Grabstätte mystischer Priester gewesen sein, die dort ihre schwarzen Messen zelebrierten. Selbst in der heutigen Zeit trauen sich die Einwohner der Dörfer nicht zu diesem Ort, der anscheinend noch nichts von seiner Mystik verloren hat. Was spielt sich noch immer dort ab, und was hat das alles mit dem Verschwinden von Onkel Jason zu tun?

In der Nähe der Grabungsstätte befindet sich ein Hünen-Grab und davor eine steinerne Statue. Von dort aus gelangt man in die Unterwelt, deren muffige Gänge von Fackeln beleuchtet sind. Nach einigen Schritten wird eine Öffnung sichtbar, die Sie wieder ans Tageslicht befördert – doch die Umgebung ist nicht mehr dieselbe. Seltsame Schlösser tauchen im Nebel auf und verlassene Städte mit hochmoderner und rätselhafter Ausstattung. Wo sind Sie gelandet und was geschieht hier? Was passiert, wenn man einen der Zaubersprüche aus dem geheimnisvollen Buch anwendet? Fragen über Fragen, und Sie müssen herausfinden, was es damit auf sich hat.

ARAZOK'S TOMB ist ein mystisches Adventure. Grafik und Sound sind gut. Die Richtungsangaben können einfach mit der Maus angegeben werden, allerdings geht es schneller, wenn man die Tastatur verwendet. Als Eingaben sind nur einfache Sätze bzw. Wortkombinationen möglich. Mehrere Anweisungen können nicht hintereinander eingegeben werden. Die Anforderungen des Spiels sind nicht sehr hoch, der Wortschatz ist einfach und man kommt ziemlich rasch voran. ARAZOK'S TOMB ist sicher kein Adventure von der Güte eines 'GUILDE OF THIEVES'. Wer allerdings ein einfaches Adventure sucht, dessen Zusammenhänge nicht zu schwierig sind, der wird mit ARAZOK'S TOMB zufrieden sein, denn Handlung und Grafik sind spannend und gut.

(mn)



Sehr schnell gelangt man in eine geheimnisvolle Welt...



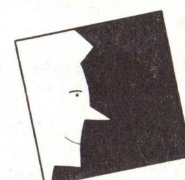
...allerdings sind deren Einwohner nicht immer nur freundlich

IDEE:

SOUND:

GRAFIK:





Anbieter: DTM, Wiesbaden
IM, 0 69 / 707 11 02

Preis: 109,–

AMIGA-KARATE

Im nächsten zu testenden Karatespiel, dem AMIGA-KARATE, ist die Hintergrundgrafik auch noch gut anzusehen, doch fehlen hier etwas die kleinen Dinge, die bei KARATE-KID II so gut gewirkt hatten.

Aber hauptsächlich kommt es ja auf das eigentliche Karate an. Die Figuren sind größer und wirken etwas schwerfällig. In den beiden ersten Leveln ändert sich nichts, das Bild bleibt das gleiche und den Gegner kann man leicht schlagen. Im dritten Level ändert sich dann das Bild, auch der Kampf wird schwieriger. Da tauchen in regelmäßigen Abständen kleine Wurfgeschosse auf, die – wenn man getroffen wird – Kraft kosten. Die Kraft wird übrigens durch Punkte über dem Bild angezeigt. Jeder Schlag, der den Gegner trifft, bekommt dieser an Punkten abgezogen.

Hat man die Levels erfolgreich hinter sich gebracht, was nicht allzu schwierig ist, weil die Gegner einfach zu schwach sind, kommt das schwierigste: Eine Reihe von Krügen hüpft hin und her, die man auf irgend eine Weise zerschlagen muß. Die Zeit dafür ist etwas knapp, und wenn nicht jeder Schlag sitzt, hat man kaum eine Chance. Schafft man es nicht, hat man verloren.

Der Zwei-Spieler-Modus weist auch eine Besonderheit auf: Während des Kampfes kommen von links oder rechts Wurfgeschosse, auf die man achten muß. Der Sound ist gut gelungen. Nach dem asiatischen Eröffnungssong hat das Spiel noch einige Geräuscheffekte zu bieten. Die Grafik ist auch recht gut, aber im Vergleich zu KARATE KID etwas langweilig. Auch hätte dem Spiel etwas Abwechslung (zum Beispiel mehr Bilder) nichts geschadet.

(AS)



Auch hier vermittelt die stimmungs volle Kulisse eine gute Atmosphäre

IDEE:

SOUND:

GRAFIK:



Anbieter: Eidersoft



KARATE-KID II

KARATE-KID II ist sozusagen das Spiel zum Film. Zwar ist der Film schon längst in den Kinos gelaufen, doch wird man gleich nach Einlegen der Diskette wieder daran erinnert, wenn das Titellied des Films 'The Glory of Love' gespielt wird. Was uns sofort auffällt, ist die hervorragende Hintergrundgrafik. Kleine Details wie zum Beispiel Vasen oder Laternen schmücken das Bild. Einmal findet der Kampf in einem Raum statt mit einem sehr schönen Wandgemälde, einmal auf einer dunklen Straße im Dämmerlicht des Mondes.

Nun zum Spiel: Daniel LaRusso kämpft gegen verschiedene Gegner, die von Kampf zu Kampf stärker werden, bis er am Ende auf den gefürchteten Chozen trifft. Während des entscheidenden Kampfes erscheint eine Trommel, deren Zweck man herausfinden muß, um den Kampf zu überstehen.

Die Bewegungen sind etwa gleich geblieben wie bei der Version für den Atari ST. Jedoch reagiert die Figur jetzt besser auf die Bewegung des Joysticks. Das Spiel geht solange, bis ein Balken aufgebraucht ist. In diesem Moment erhebt sich der Kämpfer nochmals, um gleich darauf wieder waagrecht auf dem Boden zu liegen.

Zwischen den Kämpfen gibt es aber noch einige Besonderheiten, die auch im Film vorkamen. Einmal muß Miyagi, Daniels Karatelehrer, mit zwei Eßstäbchen eine Fliege fangen. Die Übung ist nicht einfach und verlangt viel Geduld vom Spieler.

Ein anderes Mal muß Daniel einen Stapel von Eisplatten mit der Hand zerschlagen. In dieser Übung muß man mit dem Joystick Drehbewegungen ausführen, die dann immer schneller werden, um im richtigen Augenblick zuzuschlagen. Je mehr Platten durchbrechen, desto mehr Punkte erreicht man.

Die beiden Szenen fallen wieder durch ihre schöne Grafik auf. Miyagi und auch Daniel sind darin gut zu erkennen. Allerdings kann sich das Spiel etwas hinauszögern, wenn man die Fliege nicht fängt. Auch der Sound läßt nichts zu wünschen übrig, neben dem Titelsong des Films gibt es gut gelungene Geräuscheffekte. KARATE KID II ist ein wirklich gut gelungenes Karatespiel mit vielen Vorzügen.



Die Grafiken sind von bestechender, fernöstlicher Atmosphäre

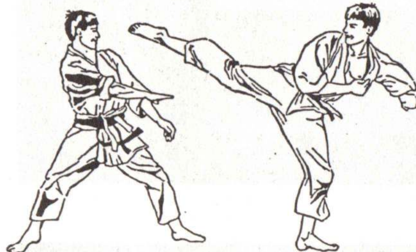
IDEE:

SOUND:

GRAFIK:



Anbieter: Microdeal



KARATE-KING

Als Kämpfer muß man versuchen, in den Palast vorzudringen. Die Stärke wird wieder durch einen Balken angezeigt. Hat man einen Kampf siegreich überstanden, kommt man dem Palast nur näher, wenn man einen Stein zerschlägt, der von rechts oder links angeflogen kommt. Man muß darauf achten, daß man nicht zu früh oder zu spät schlägt. Im zweiten Level sind es dann zwei Steine, im dritten drei, und so weiter. Aber man darf sich das Weiterkommen nicht so leicht vorstellen!

Im Vergleich zu den beiden anderen Karatespielen ist KARATE-KING das schnellste – meiner Meinung nach sogar etwas zu schnell. Als Anfänger kommt man über den zweiten Level normalerweise nicht hinaus. Die Bewegungen, die man auf die Spieler ausübt, sind weitgehend die gleichen wie bei den anderen Karatespielen, nur etwas schneller. Ist man in der glücklichen Lage, bis zum sechsten Level zu kommen, muß der Spieler Pfeilen und anderen Dingen ausweichen. Die Grafik ist nicht so gut gelungen, alles sieht etwas verschwommen aus. Auch muß man schon ein Karate-Profi sein, um jemals in den Palast zu gelangen, deshalb ist für einen Karate-Anfänger wie für einen, der auch Wert auf gute Grafik legt, das Spiel nicht zu empfehlen.



Grafik und Landschaft sind etwas karg

IDEE:

SOUND:

GRAFIK:



Anbieter: Kingsoft
Preis: 49,95

THE KING OF CHICAGO

Chicago in den dreißiger Jahren: Prohibition, Machtkämpfe, Bandenkriminalität und Gesetzlosigkeit. Nur die Härtesten unter den Harten überleben. Capone war der Boss der Bosse, aber Capone ist jenseits des Flusses und hikago ist jenseits jeder Kontrolle. Gewalt, Schmeicheleien, Bündnisse und Verrat sind die Schlüssel zur Macht – aber man muß wissen, wann und wie man sie einsetzt.

THE KING OF CHICAGO

Chicago in den dreißiger Jahren: Prohibition, Machtkämpfe, Bandenkriminalität und Gesetzlosigkeit. Nur die Härtesten unter den Harten überleben. Capone war der Boss der Bosse, aber Capone ist jenseits des Flusses und Chikago ist jenseits jeder Kontrolle. Gewalt, Schmeicheleien, Bündnisse und Verrat sind die Schlüssel zur Macht – aber man muß wissen, wann und wie man sie einsetzt.

Das Szenario:

THE KING OF CHICAGO ist ein interakiver Film, der den Spieler in der Zeit zurückführt und ihm die Chance gibt, die Geschichte zu ändern. Die Story beginnt mit dem Sturz des größten Gangsterkönigs aller Zeiten. Al Capone wird von der Regierung zu einem elfjährigen Urlaub in der Festung von Alcatraz eingeladen. Seine Bande – die Southside Gang – übernimmt sein Vertrauter, der gefährliche Toni Santucci.

Die Hauptrolle jedoch spielt Pinky Callahan, ein Bandenführer der Northside Gang. Pinky ist des blutigen Bürgerkriegs müde. Er will diese windigen Typen aus der Südstadt in den Michigansee treiben und Chicago alleine regieren. Er will der König von Chicago werden, und nichts kann ihn aufhalten.

Zu seinem Pech ist Pinky aber nicht die Nummer eins im Norden der Stadt. Zwei andere Gangsterbosse stehen noch über ihm. Da ist zuerst der Grosse Alte Mann, der schon zu Capones Zeiten eine große Nummer war. Das zweite Hindernis ist Ben, die rechte Hand des Alten. Wenn der Chef plötzlich ablebt, soll er dessen Platz einnehmen. Aber Pinky hat die berechtigte Hoffnung, daß Ben mit den richtigen Argumenten überredet werden kann, sich in diesen Sessel erst gar nicht zu setzen.

Die Mitspieler:

Die Braut des Gangsters ist nicht sein Gewehr, sondern aus Fleisch und Blut: Pinkys Braut Lola ist sogar besonders heißblütig. Wer sie schlecht behandelt, bekommt ihren Zorn zu spüren.





Tom Malone fühlt sich als echter Polizist oder auch als menschlicher Staubsauger – vollgesogen mit dem Geist der Humanität und Pflichterfüllung.

Den nervösesten Zeigefinger in dieser illustren Runde besitzt Guido, Tony Santuccis rechte Hand. Seine eigene rechte Hand zuckt bei jeder Gelegenheit. Er favorisiert stets endgültige Lösungen.

Marcello Merinotti, der Industriemagnat und Devisenschieber bleibt im Hintergrund – so gut, daß er eigentlich gar nicht zu finden ist.

Der Film:

Die Handlung beginnt 1931. Sie ist zeitlich begrenzt. Den Abschluß bildet das Treffen der amerikanischen Gangsterbosse 1934 in New York. Dort wird das nationale amerikanische Syndikat gegründet. Die Mord Inc. wurde gebildet, um die Macht zu zentralisieren und die Kriege der kleinen rivalisierenden Banden zu beenden.

Chicago wurde als zu barbarisch eingestuft, um ein stimmberechtigtes Mitglied dieser Gesellschaft zu werden.

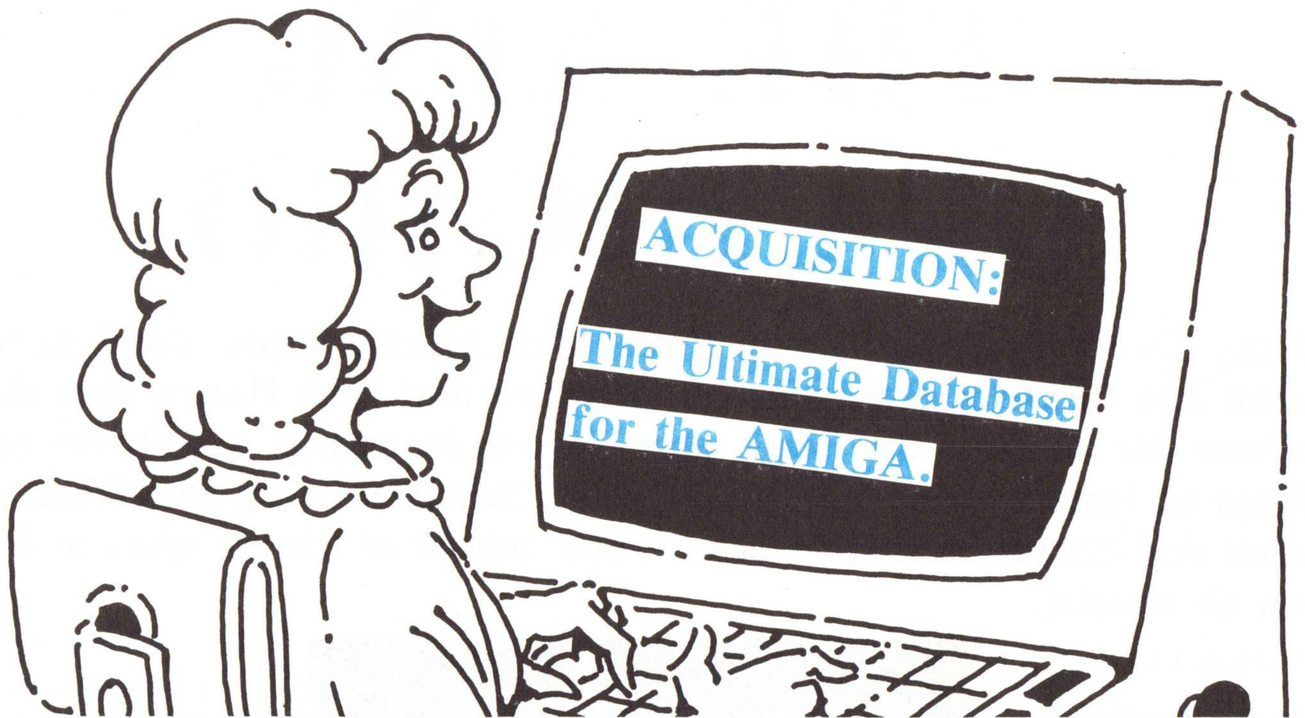
Dies ist Ihre Chance, die Geschichte umzuschreiben. Sie haben drei Jahre Zeit, Chicago in die Hand zu bekommen und die Bandenkriege zu stoppen. Im Erfolgsfall werden Sie von der ehrenwerten Gesellschaft in New York als Mitglied akzeptiert und können die Stadt nun getrost in die Tasche stecken, denn Sie sind der KING OF CHICAGO!

Spielstrategie:

KING OF CHICAGO ist ein Strategiespiel. Jedesmal, wenn der Spieler eine Entscheidung trifft, verändert er damit sowohl die Persönlichkeit des Charakters, den er repräsentiert, als auch die Persönlichkeit der Mitspieler.

Es gibt zahlreiche Chancen, KING OF CHICAGO zu gewinnen. Es gibt aber auch zahllose Möglichkeiten, baden zu gehen – im Michigansee.

(cpl)



So bezeichnet die Firma TAURUS ihr Datenbankprogramm für den Amiga. Was sofort ins Auge fällt, ist die hervorragende Aufmachung der Programmbeschreibung. Geliefert wird eine stabile graue Hülle, in der sich das Handbuch, zwei Disketten und der Dongle für den Kopierschutz befinden.

Das Handbuch (in englisch) ist sauber gegliedert und führt den Anwender Schritt für Schritt in die Benutzung des Programms ein. Um mit ACQUISITION arbeiten zu können, benötigt man einen Speicher von mindestens 512K RAM und 2 Floppy-Disks oder eine Festplatte.

Das Programm weist Leistungsdaten auf, die glauben lassen, man habe es hier mit einem Programm für eine Großrechenanlage zu tun. So ist es z. B. in der Lage, 10 000 Felder in einem Satz zu verwalten. Eine Datei kann bis zu 1 Gigabyte groß sein. Damit stellt ACQUISITION so ziemlich alle anderen Programme dieser Art für PCs in den Schatten.

Trotz dieser enormen Größen ist die Handhabung des Programms auch für den Laien sehr einfach. In den schön gestalteten Grafikmenues kann man die einzelnen Abschnitte zum Erstellen und Bearbeiten von Dateien mit Icons anwählen. Es ist also nicht nötig, eine Programmiersprache zu erlernen, um sich eine Datenbank anzulegen.

ACQUISITION wurde in einer eigenen Sprache „Acom“ (ähnlich BASIC) programmiert.

Bevor man mit ACQUISITION arbeiten kann, muß das Menue SETTINGS aufgerufen werden. Hier erfolgt eine Anpassung des Programms an die eigene Gerätekonfiguration. Danach kann man schon beginnen, sich eine Datei zu erstellen und damit zu arbeiten. Das geschieht in dem Programmteil CREATING.

Mit PASTING erstellt man sich Bildschirmmasken für die Daten-Ein- und Ausgabe. Hier kann man auch festlegen, welche Daten einer Datei angezeigt werden sollen.

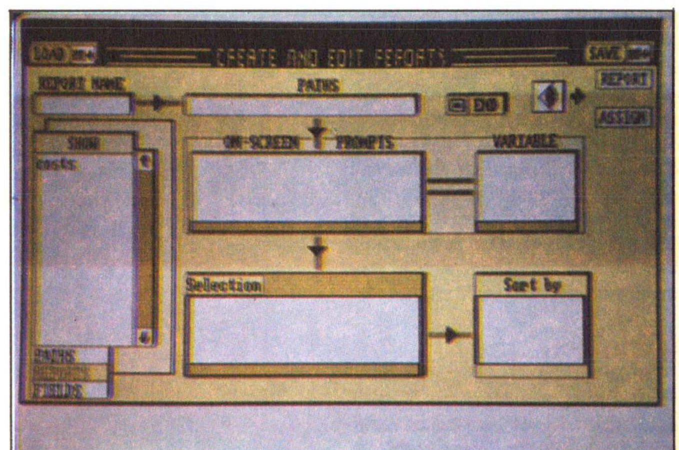
PASTING enthält auch ein kleines Grafikprogramm, um die einzelnen Anzeigefenster zu malen. Um festzustellen, wann ein Datenzugriff erfolgt ist, kann man sich das Datum und die Uhrzeit in die Datei eintragen lassen. Dabei hilft ein kleines Programm, das über ein Icon angeklickt werden kann. Bei ACQUISITION ist es nicht erforderlich, sämtliche benötigten Daten in eine Datei zu packen. Nehmen wir an, Sie möchten Bücher verwalten. Dann müssen Sie zu einem Autor nicht alle seine Bücher eintragen. Dafür legen Sie sich eine gesonderte Datei an, die Sie mit BRIDGING wieder mit der Autorendatei verbinden können. Es wird dazu ein Pfad angegeben, der auf die zusammengehörenden Daten weist. Mit LINKING werden die Daten dann zusammengeführt.

FILING ist das Programmteil, mit dem man Datensätze editieren kann.

ACQUISITION ist ein Programm, das sich durch seine einfache Bedienung und die Möglichkeit, große Datenmengen zu verwalten, auszeichnet. Es ist sehr flexibel, da man es auch nach seinen Wünschen durch programmieren in „Acom“ (Befehlssyntax wird im Handbuch erklärt) erweitern kann. ACQUISITION enthält noch einige nützliche Zusatzfunktionen wie Taschenrechner, Uhr, Kalender und ein kleines Textprogramm. ACQUISITION läßt kaum Wünsche beim Anwender offen und ist ein empfehlenswertes Programm.

wosp

Bezugsquelle:
Softline, Oberkirch
Preis: ca. 639,– DM





Online im Duett

Neue Wege der Datenfernübertragung mit Aegis DIGA!

Für Amiga-Benutzer ist die Softwareschmiede Aegis Development im südkalifornischen Santa Monica längst kein unbekannter Name mehr. Mit dem Animationsprogramm Animator/Image und dem Musikprogramm Sonix zeigte das gewiefte Programmiererteam bereits, was sich alles mit einem Amiga anstellen läßt. Der gute Ruf aller Aegis-Produkte eilt auch ihrem neuesten Programm DIGA! voraus. In den USA, dem Mekka der Datenfernübertragung (kurz DFÜ genannt) gab es viel Lob von den Fachzeitschriften. Grund genug, dieses brandaktuelle Produkt näher unter die Lupe zu nehmen.

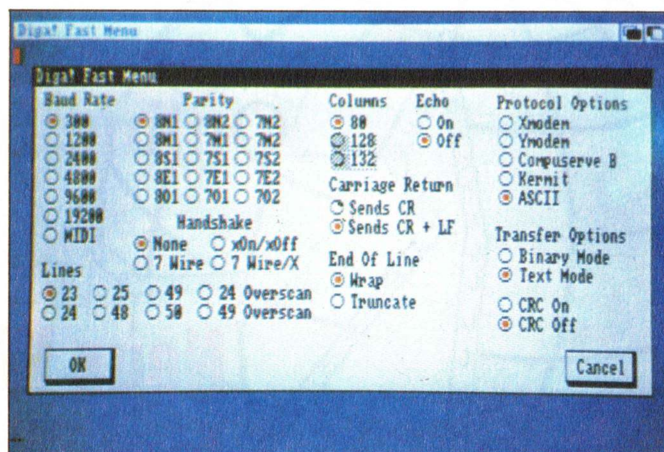
Bereits nach dem Programmstart und dem Anwählen des ersten Pulldownmenüs wird klar: DIGA! bietet mehr als vergleichbare Konkurrenten. Im Menü Project läßt sich ein sogenanntes Fast Menu aufrufen. Hier kann der DIGA-Benutzer in einer Requester-Box die Grundeinstellung für die Datenübertragung und Bildschirmausgabe einstellen. Die Baud-Rate kann von 300 bis 19 200 Bits pro Sekunde mit der Maustaste eingestellt werden. Zusätzlich gibt es noch eine MIDI-Übertragungsgeschwindigkeit. Hier kann man Daten von zwei direkt über die serielle Schnittstelle verbundenen Rechnern mit einer Geschwindigkeit von 31 500 Bits pro Sekunde übertragen – und dies alles bei höchster Datensicherheit. Ein C64 kommt da gerade auf 1200 Baud.

Für korrekte Datenübertragung sorgt nicht zuletzt das gesetzte Paritätsbit. Dieses Bit wird immer dann gesetzt, wenn eine ungerade Anzahl von Datenbits gesendet wird. Ist 0bereits eine gerade Anzahl von Datenbits gesetzt, so wird das Paritätsbit gelöscht. Auf diese Weise wird durch Quersummenbildung der Datenbits erst eine Fehlerüberprüfung der übertragenen Information möglich. Als zusätzliche Sicherheit sind noch Start- und Stopbits vorhanden. Anhand des Startbits erkennt der Empfangscomputer, daß jetzt ein Byte übermittelt wird. Damit der Empfangscomputer erkennt, wann ein Byte zu Ende ist, sendet man noch ein oder zwei Stoppbits. DIGA! bietet hier insgesamt 15 verschiedene Möglichkeiten der Paritätseinstellung. Neben anderen Grundeinstellungen wie Handshake oder Carriage Return gibt es im Fast Menü noch verschiedene Übertragungsprotokolle. Neben dem geläufigen XModem, Kermit oder ASCII-Protokoll findet man auch YModem und Compuserve-B. Diese verschiedenen Protokolle werden in erste Linie verwendet, wenn es um die exakte Übertragung von ganzen Programmfiles geht. Dabei müssen beide Rechner das gleiche Protokoll verwenden. Wer über genügend Kleingeld für die Telefonrechnung verfügt, kann somit auch von der amerikanischen Mailbox Compuserve Public-Domain-Programme in seinen Rechner laden. Und mit Doubletalk gibt es noch ein spezielles Übertragungsprotokoll, das es allen DIGA-Besitzern ermöglicht, gleichzeitig Pro-



grammfiles zu senden oder zu empfangen, während man mit einem anderen DIGA-Benutzer Mitteilungen austauscht. Doubletalk benutzt bei dieser Art der Datenübertragung ein eigenes Paketsystem und spezielle Fehlerüberprüfungsroutinen. Somit ist auch der Filetransfer über große Entfernungen bei hohen Baudraten gewährleistet. Der Bildschirm wird dabei in drei Fenster aufgeteilt. Im obersten Window stehen dabei die sogenannten Eingangsgespräche. Im Mittelfenster werden Mitteilungen oder Datenfiles an den Gesprächspartner abgeschickt. Das Transfer-Status-Fenster informiert ständig über aufgetretene Übertragungsfehler, welcher Datenblock gerade gesendet oder empfangen wird und wieviel Prozent der Datei bereits übertragen wurde. Im Fastmenü kann man dazu noch festlegen, ob man ASCII- oder Binärdateien übertragen will. Auf dem Bildschirm kann man sich die Daten dann mit 80, 128 oder 132 Spalten anzeigen lassen, wobei letztere Einstellung wirklich nur bei sehr guten Augen oder einem hervorragendem Monitor zu empfehlen ist.

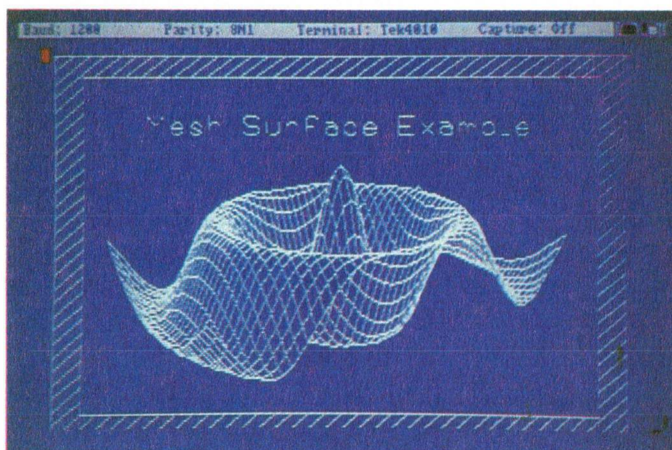
Die Zeilenanzahl läßt sich flexibel von 23 bis 50 Zeilen anzeigen. Die Aegis-Leute haben hier mit Overscan gearbeitet. Das heißt, bei einer 49- oder 50-zeiligen Darstellungsart wird der Bildschirm auch bei europäischen PAL-Amigas vollständig genutzt. Zusätzlich kann DIGA! noch verschiedene Bildschirmterminals emulieren. Hier kann man zwischen TTY, ANSI, VT100, VT52 und Tektronix 4010/14 wählen. Dabei ist die Tektronix-Emulation wohl die spektakulärste, da sie den Grafikmodus dieses Terminals unterstützt. Vom einem Tektronix-Terminal übermittelte Grafiken können mit Hilfe dieser Emulation direkt auf dem Bildschirm angezeigt werden. Bildschirmausschnitte kann man mit Hilfe der Zoom-In-Funktion schrittweise vergrößern. Zusätzlich befindet sich auf der DIGA-Programmdiskette das Konvertierungsprogramm TektoDraw. Tektronix-Grafikfiles können mit Hilfe dieses nützlichen Tools in das Fileformat das CAD-Programms Aegis DRAW umgewandelt werden. Dort lassen sich dann diese Grafiken editieren und auf einem Plotter ausgeben.



Schnelle Parameterkontrolle mit dem FAST MENÜ

Trotz vielfältiger Möglichkeiten bei der Terminalwahl und Bildschirmdarstellung ist DIGA! nicht BTX-fähig. Es ist zu hoffen, daß diese Option in späteren Programmversionen berücksichtigt wird. Daß Aegis-DIGA mehr für den amerikanischen Markt konzipiert wurde, stellt man auch im Phone Menü fest. Alle Voreinstellungen im Programm sind für ein Hayes-kompatibles Modem. Und wer nicht gerade mit einem Modem ohne FTZ-Zulassung oder gar einer Selbstbaukonstruktion erwischt werden möchte, muß hierzulande sehr tief in das Portemonnaie greifen. Und Modeme gibt es nur von der Post. Von anderen Herstellern wurden bis jetzt keine Modeme zugelassen. So muß auch der DIGA-Anwender auf so angenehme Sachen wie automatisches Wählen, Wahlwiederholung und selbständiges Abbrechen der Verbindung verzichten. Genauso kann man das integrierte Telefonbuch zwar zur Speicherung von Adressen und Telefonnummern verwenden, das Anwählen der Mailbox bleibt bei Verwendung eines Akustikkopplers selbst überlassen. Sollte man dennoch im Besitz eines Modems sein, kann man im Menüpunkt Modem Setup nd die nötigen Ansteuerungssequenzen anpassen. Mit Remote Access haben die DIGA-Entwickler sogar eine Art Mini-Mailbox als Programmodul eingefügt. Diese Remote-Funktion erlaubt es einem Benutzer, von einem anderen Rechner via Telefonleitung DIGA! zu benutzen. DIGA! wartet dann auf hereinkommende Telefongespräche und antwortet selbständig. Der Anrufer wird mit einem selbst erstellten Textfile begrüßt und dann nach einem Paßwort gefragt. Hier gibt es zwei Paßworte. Das User-Paßwort verschafft nur Zutritt zu den Grundfunktionen. Das Haupt-Paßwort jedoch läßt auch Funktionen wie Files löschen und kopieren zu.

Überschreitet der Benutzer die frei definierbare Online-Zeit, oder findet für zwei Minuten keine Übertragung mehr statt, beendet DIGA! die Verbindung. Mit der aktivierten Funktion Caller Log wird dann sogar Datum und Zeit des Anrufers gespeichert. Selbst besondere Vorkommnisse wie falsche Eingabe eines Paßwortes oder vorzeitiges Abbrechen



Auch Grafiken lassen sich mit DIGA! übertragen und empfangen.

der Verbindung werden notiert. Leider – wie oben bereits erwähnt – funktionieren alle diese Funktionen nur für die Besitzer eines Modems. Auch DIGAs DFÜ-Programmiersprache profitiert von diesem Umstand nicht. Mit Script Language kann sich jeder eigene kleine Batch-Dateien zur Modemansteuerung schreiben. Nützliche Befehle wie Baud oder Exec für das Einstellen der Baudrate oder Aufrufen eines AmigaDos-Programmes werden hier durch Schleifen und Sprungbefehle ergänzt. Nutzlos bleiben jedoch Befehle wie etwa Dial oder Hangup für automatisches Wählen oder Beenden der Verbindung.

Eine überaus nützliche Funktion ist das Definieren der Funktionstasten mit bis zu 80 Buchstaben langen Texten. Durch die Kombinationen SHIFT-, ALT-, LEFT AMIGA-Taste mit den 10 Funktionstasten ergeben sich insgesamt 50 frei programmierbare Tasten. Auf diese Weise erspart der Anwender sich die oft lästige Tipperei bei langen Controlsequenzen.

Bevor man jedoch die nächstgelegene Mailbox anklingelt, sollte noch die Buffergröße für die Empfangsdaten festgelegt werden. Hier kann man zwischen 4 KByte – entspricht ungefähr 4000 Buchstaben – und 512 KByte wählen. Bei einem 512 KByte Amiga stehen dem DIGA-Benutzer immerhin noch 168 KByte für den Buffer zu Verfügung. Im Capture-Modus kann man dann alle ankommenden Daten im Buffer zwischenspeichern. Ist der Loop-Modus in Aktion, wird bei vollem Buffer die älteste darin befindliche Information wieder überschrieben. Alle diese Grundeinstellungen müssen selbstverständlich nicht jedesmal wiederholt werden. Im Configuration-Menü können alle Voreinstellungen des Programms gespeichert werden. Hier können sogar mehrere unterschiedliche Konfigurationsfiles erstellt und bei Bedarf aufgerufen werden.

Fazit:

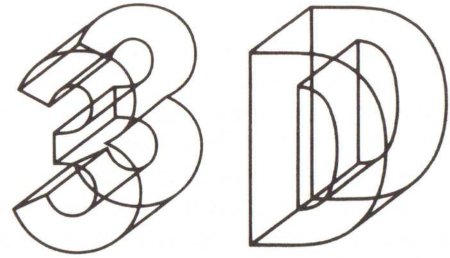
Sieht man einmal von der fehlenden BTX-Funktion und den für Besitzer von Akustikkopplern nicht benutzbaren Funktionen ab, so bleibt DIGA! doch ein DFÜ-Programm mit zahlreichen Optionen und vielfältigen Möglichkeiten. Das 94seitige Handbuch unterstützt durch viele Beispiele und Erklärungen den Anwender bei seinen Problemen. Den Vergleich mit anderen DFÜ-Programmen auf dem Amiga muß DIGA! keinesfalls scheuen. Es kann deshalb für alle Arten der Datenübertragung empfohlen werden.

(ROS)

PLUS/MINUS KASTEN:

- \+ komfortables Menühandling
- \+ eingebaute DFÜ-Programmiersprache
- \+ alle Standardübertragungsprotokolle
- \+ eingebaute MiniMailbox
- \+ eingebautes Telefonbuch
- \+ frei definierbare Funktionstasten
- \+ Darstellung von Grafikbildern

- – nicht BTX-fähig
- – viele Befehle nur für Modem

VIDEO SCAPE 

Der 3D-Animator aus dem Hause Aegis Development Sparta Inc.

Der Aegis Animator sorgte vor einiger Zeit für Aufregung in der Öffentlichkeit. Das Programm, das durch leichte Handhabung und atemberaubende Möglichkeiten besticht, konnte auch den verwöhntesten Amiga-Freak in Begeisterung versetzen.

Und nun ein neuer Animator – dreidimensional. Meine Erwartungen an dieses Programm waren hoch. Kein Wunder, denn schon der erste Animator war faszinierend. Das Softwarepaket umfasst drei Disketten und ein ausführliches User-Manual. Das ist zwar in englischer Sprache, aber auch für wenig Begabte (wie mich) leicht verständlich. Die Hauptdiskette enthält die Programme VideoScape 3D, Designer 3D, 3D Hints, EGG, OCT und einige Schubladen. Pictures heißt die eine der beiden anderen Disketten. Damit ist ihr Inhalt schon beschrieben: Beim Öffnen erscheint in einem winzigen Fenster nur die Schublade „pic". Das heißhungrige Anklicken wird gleich enttäuscht. Es erscheint ein ebensokleines Fenster, das leer ist. Doch die Diskette ist nicht leer, sondern die Bilder-Icons wurden kurzerhand weggelassen. Auf die Bilder kann man also nur über das CLI oder über den Animator (oder über jedes andere IFF-kompatible Programm wie DPaint II, Aegis Images etc.) zugreifen. Die zweite Diskette mit dem Titel „Objects" lässt ebenfalls nur leere Schubladen zum Vorschein kommen. Diese Diskette kann man als eine Art Datendiskette bezeichnen. Sie ist dafür gedacht, fertige Videos, Objekte, Kameraeinstellungen etc. abzuspeichern. Leer ist allerdings auch diese Diskette nicht. Sie enthält einige Demos und einige für Anfänger gedachte Objekte, Motions und sonstige Daten, die man braucht, um ein gutes Video zu machen. Nun zur Hauptdiskette: Beim Anklicken dieser Diskette erscheinen die erwähnten Icons. Das Softwarepaket VideoScape 3D besteht in erster Linie aus zwei Programmen. Sie heißen „Designer 3D" und „VideoScape 3D". Designer 3D ist dazu gedacht, 3D-Objekte zu zeichnen und sie geschickt zu editieren. Dieses Programm ist keines der herkömmlichen Zeichenprogramme für den Amiga. Andere Zeichenprogramme beschränken sich auch nur auf zwei Dimensionen. Vor dem Laden des Programmes sollte man allerdings die 3D-Hints gelesen haben. Dieses Programm erklärt (in Englisch) das Programm Designer 3D. Wer kein gutes Gedächtnis hat und einen Drucker besitzt, kann sich diesen Text auch ausdrucken lassen. Und das geschieht so: „type D3Dhints.txt par:". Nun wird eine ungefähr vier Seiten lange, englische Anleitung ausgedruckt. Das Programm Designer 3D gliedert sich in zwei Teile. Zum einen der „Object Editor", zum anderen der „Action Editor". Der Object-Editor ist, wie der Name sagt, dazu gedacht, dreidimensionale Objekte zu entwerfen. Leider ist dieser Editor noch schlechter ausgefallen als die Zeichenroutinen bei dem 2D-Animator von Aegis.



Einfache Menüsteuerung aller Funktionen

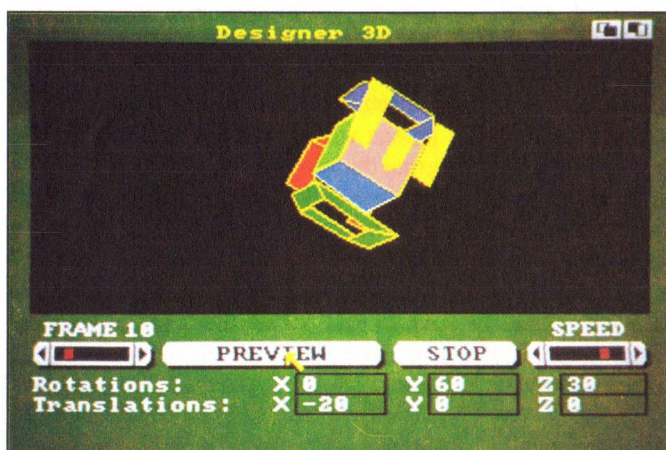
Für die Ansprüche, die an ihn gestellt werden, reicht er aber aus. Um ein dreidimensionales Bild zu entwerfen, braucht man auch drei Seiten. Im technischen Zeichnen werden diese Ansichten die „Drei-Tafel-Projektion" genannt. Um ein Objekt zu zeichnen, wird einfach ein Punkt in eine der drei Ansichten gesetzt, danach ein zweiter Punkt, der mit dem ersten automatisch verbunden wird. Jetzt ein Dritter, und so weiter. Zwischen jedem Setzen eines Punktes ist ein Punktzähler nach oben zu klicken. Dieser Punktzähler hat die Form eines Rollbalkens und wird genauso bedient. Das gewünschte Objekt muss aber nicht aus einem Polygon bestehen, das wäre etwas umständlich. Ist das Polygon, hier in der Bedeutung als ein Teil des fertigen Objekts, fertig, kann ein Polygonzähler hochgesetzt werden. Dieser funktioniert genau wie der Punktzähler. Beim Umschalten auf ein neues Polygon er-

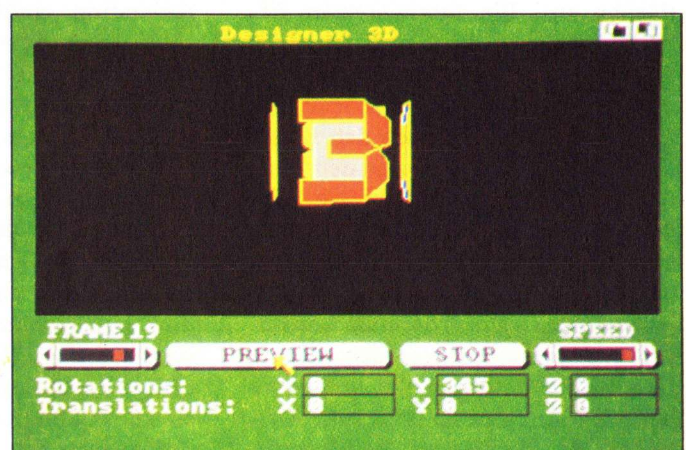
scheint das aktuelle Polygon in einer anderen Farbe. Da bei sehr vielen Polygonen die einzelenen Ansichten sehr verwirrend werden, gibt es einen Pull-Down-Menüpunkt, der alle Polygone bis auf das aktuelle ausschaltet. Nun sind nur noch die Punkte der anderen Polygone zu sehen. Die Punkte müssen allerdings nicht unbedingt mit der Maus gesetzt werden. Es gibt auch die Möglichkeit, die Koordinaten der Punkte anzugeben, die dann auch prompt gesetzt werden. Hierfür stehen Eingabebereiche auf dem Screen zur Verfügung. Das kann ganz nützlich sein, wenn ein Polygon sehr genau gesetzt werden muß. Allerdings muß man hierfür schon einiges Wissen aus der Geometrie mitbringen. Für mich schied diese sicher sehr sinnvolle Möglichkeit schon von vorneherein aus.

Die Editiermöglickeiten beschränken sich auf die Funktionen, die letzte Aktion zurückzunehmen (Undo) und ein Polygon zu löschen. Ein Polygon muss nicht in allen drei Ansichten gezeichnet werden. Eine Ansicht genügt völlig, denn die anderen Ansichten werden perspektivisch richtig gezeichnet. Menü-Punkte wie Load und Save fehlen natürlich nicht. Will man ein Objekt im Programm Video-Scape 3D verwenden, denn das Designer 3D Format versteht das „dumme" VideoScape Programm nicht, hat man die Möglichkeit, es unter diesem Format abzuspeichern. Ist das gewünschte Objekt fertig und man möchte es auch mal ein wenig durch die Gegend „schleudern", schaltet man in den Action Editor um. Der Action Editor Der Action Editor ist ein kleines, allerdings sehr leistungsstarkes 3D-Animationsprogramm. Hier sind der Phantasie keine Grenzen gesetzt. In alle erdenklichen Richtungen kann unser Objekt gedreht werden. Der Bewegungsablauf, der in 24 Frames unterteilt ist, wird mit Anklicken des „PREVIEW" Kastens gestartet. Schaltet man zum ersten Mal in den Action-Editor um, sind die Koordinaten so eingestellt, daß sich das Objekt um die X-Achse dreht. Der erste Durchlauf der 24 Frames ist enttäuschend, denn er ist sehr langsam. Doch sind alle 24 Frames einmal gelaufen, wird die Animation schnell und fliessend. Die Erklärung liegt auf der Hand: Beim ersten Durchlauf werden die einzelnen Bilder neu berechnet und die Werte im RAM abgelegt. Bei allen anderen Durchläufen werden diese Werte einfach nur aus dem Speicher geholt. Ein Rollbalken zum Einstellen der Geschwindigkeit fehlt auch nicht. Bei Höchstgeschwindigkeit bleibt manchen, auch mir, der Mund offen stehen. Daß der Amiga ein sehr schneller Computer ist wußte ich, aber so schnell? Das Bewegen des Objekts erfolgt durch Eingabe der Koordinaten. Mit einiger Übung bekommt man den „Dreh" schnell raus. Also nicht verzweifeln, wenn das Objekt am Anfang nicht das macht, was man will! Hat man ein Objekt gezeichnet und ist mit allen Ansichten zufrieden, speichert man es im Video-Scape-Format ab.

VideoScape 3D, der eigentliche Animator

Nach dem Anklicken des Video-Scape-Symbols erscheint ein ganz nettes Bild mit dem Namen des Autors, Allen Hastings, und gleich danach das Control-Window. Man kann dieses Window auch als Computermischpult bezeichnen. Das Control-Window ist in drei Teile gegliedert. Links oben befindet sich das „Object Description"-Fenster. In diesem Fenster wird das Objekt, das geladen ist, genau beschrieben. Die Anzahl der Punkte und die Anzahl der Polygone werden angezeigt. Nach dem Anklicken des Kastens „Load Object" erscheint ein neues Fenster, das den Inhalt der Datei „geo" zeigt. In der Datei „geo" sollten alle Objekte gespeichert werden, da man sonst den Überblick verliert. Der Name des gewünschten Objektes wird angeklickt und erscheint im Filekasten. Nun wird „OK" gewählt, und das Fenster macht „dicht". Sofort erscheint ein neues Fenster, das mit „Object Motion Storage Requester" betitelt ist. In diesem Motion File sind die Positionen und Bewegungen des Objekts beschrieben. Dieses File ist praktisch eine fertige Animationssequenz. Haben wir kein solches File, können wir auch keinen Namen angeben; wir lassen das File-Fenster daher leer und gehen auf „OK". Dieses Fenster bietet allerdings noch einen ganz ungewohnten Selektorpunkt mit dem Titel „Manual". Klickt man diesen Punkt an, erscheint ein Fenster, in dem man die Startposition und die Endposition des Objekts angeben kann. Sind zum Beispiel die X-Werte unterschiedlich, dreht das Objekt um die X-Achse. Nach erfolgreichem Durchwandern des „Fenster-Labyrinths" wird das Objekt geladen. Prompt werden die Anzahlen der Punkte und der Polygone angezeigt. Das User-Manual hat also nicht gelogen. Im Object-Description-Fenster befindet sich noch eine „Yes/No"-



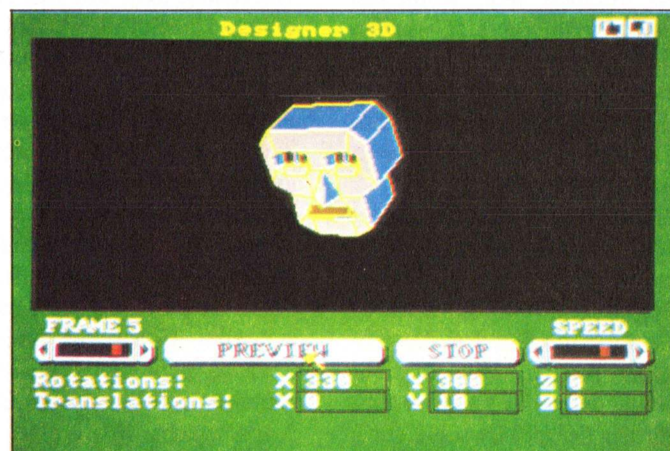


Auswahlmöglichkeit mit dem Titel „Metamorph last two objects". Zwar ist hier „No" eingestellt, aber es lohnt sich, auf „Yes" umzuschalten. Diese Möglichkeit, ein Objekt in ein anderes zu verwandeln, ist schon vom 2D-Animator her bekannt, allerdings nicht zu vergleichen. Voraussetzung für diese „Computer-Evulotion" ist allerdings, daß die letzen beiden Objekte dieselbe Anzahl von Punkten hatten. Danach verwandelt sich das vorletzte Objekt in das letzte. Unterhalb des „Object Description"-Fensters befindet sich das „Camera Motion"-Fenster mit drei „Knöpfen". Der Punkt „Load Motion" aktiviert ein „Load Fenster". Dort kann man ein Motion File laden, falls man es beim Laden des Objekts vergessen hat.

Ein weiterer Punkt ist der Punkt „Change Zoom". Alle Hobbyfotografen wissen sofort, was damit gemeint ist. Hier kann man die Kamera so einstellen, ob sie nahe (hoher Zoomwert) oder weiter weg (niedriger Zoomwert) am Objekt ist. Eingestellt ist ein Wert von 1.6. Das „Camera Motion"-Fenster enthält noch eine Ja/Nein-Abfrage, mit der man einstellen kann, ob zwischen jedem Bild eine Pause erwünscht wird. Zusätzlich gibt es zwei Zähler, die „Tweens" und „Frames" zählen. Als Tween bezeichnet man den Zeitabstand zwischen zwei Frames.

Das dritte und letzte Window im Control-Window nennt sich „Viewing Options". In diesem Fenster werden die Screenfunktionen eingestellt. Man kann nach dem Ja/Nein-Prinzip zwischen „Wire" und „Solid" wählen. „Wire" ist eine Art Outlined. Hier werden nur die Linien der Polygone gezeichnet. Solid füllt die Polygonflächen aus. Ein faszinierender Effekt bei Solid ist die Simulation einer Lichtquelle. Nach dem Umschalten von Wire nach Solid erscheint ein Eingabebereich, der nach den Koordinaten der Lichtquelle und dem Einfallswinkel fragt. Die Objekte werden später von einer „Computersonne" angestrahlt. Die sich daraus ergebenden Farbschattierungen des Objekts werden vom Programm berechnet und gezeigt. Unter den Kästen zum Einstellen von Solid und Wire befinden sich vier Knöpfe zum Auswählen der Auflösung (Resolution). Angeboten werden folgende Auflösungsmöglichkeiten: 352 X 220, 352 X 440, 704 X 220 und 704 X 220. Die Hintergrundgrafiken müssen der Auflösung entsprechen, da sonst ein Chaos auf dem Bildschirm entsteht. Der Screen, auf dem die Objekte sich bewegen sollen, ist in zwei Hälften unterteilt. Der obere Teil nennt sich Sky, der untere Ground. Die Farben können auf dem Control-Window recht einfach eingestellt werden. Mit Pfeilen können 16 Farben für Himmel und Erde bestimmt werden. Der Horizont, der so entsteht, ändert sich natürlich mit dem Einfallswinkel der Kamera. Schließlich sind noch zwei Knöpfe zum Laden der „Foregrounds" und „Backgrounds". Das Laden erfolgt nach Schema F (wie bei dem Laden von Objekten, Motions etc.). Mit Pull-Down-Menüs ist das Control-Window geradezu ärmlich ausgestattet. Das obligatorische „About" fehlt natürlich nicht. Auch ein „Quit" ist vorhanden. Interessant und wichtig sind die Menüpunkte „Begin Recording" und „End Recording". Nachdem alle nötigen Dinge für das Video geladen sind und die Animation endlich gestartet wird, sollte man allerdings diese Werte speichern. Am unteren Bildschirm befinden sich die dazu nötigen Punkte (Save Settings, Load Settings).

Warum diese Umständlichkeit? Wenn man nach der Animation wieder zurück in das Control-Window kommt, sind sämtliche Einstellungen verschwunden und das Control-Window erscheint in seinem ursprünglichen Zustand. Nach Anwählen des „Begin Recording"-Punktes wird nach einem Animationsfile gefragt, das den ganzen Schweiß unserer Arbeit spiegelt. Ist das File definiert, kann nun endlich die Animation gestartet werden. Noch ein Hinweis: Besitzer eines zweiten Laufwerks sollten das Animationsfile auf diesem Laufwerk mit einer leeren Diskette speichern. „Discjockeys" mit nur einem Laufwerk legen nach dem Starten eine formatierte Diskette ein. Das Animationsfile benötigt extrem viel Speicherplatz auf der Diskette. Abhängig ist dies natürlich von der Anzahl der Polygone, die animiert werden sollen. 200 Frames können verwaltet werden. Ein großes Objekt im „Wire Modus" braucht zum Beispiel, bei 100 von 200 möglichen Frames, über 400 KByte!



Jetzt aber endlich zum interessanten Teil, dem Animieren. Am unteren Bildschirmrand befindet sich der Punkt „Begin Animation". Nach dem Anklicken wird der Bildschirm schwarz und der Text „Working" ist zu lesen. Wenn die Arbeit getan ist, erscheint unser Objekt. Mit dem Ziffer-block auf der rechten Seite der Amigatastatur kann dieses Objekt nun bewegt werden. Alle Richtungen sind möglich. Auch eine Zoomfunktion ist vorhanden. Ein schöner Effekt ist es, durch das Objekt, sofern es im „Wire"-Modus ist, hindurchzufahren. Falls die Tastenbelegung vergessen wurde, kann durch die Betätigung der rechten Maustaste ein Quickreminder in Form eines Fensters auf dem Bildschirm eingeschaltet werden. Die linke Maustaste gibt Informationen über das Objekt wieder. Koordinaten und Anzahl der Frames werden hier gezeigt. Der Dezimalpunkt auf dem Ziffernblock ermöglicht es, das aktuelle Bild als IFF-Bild abzuspeichern. Hat man die Kameraeinstel-

lung durch das Drücken einer Taste geändert, verschwindet das Objekt, um kurz danach in der neuen Form zu erscheinen. Dieser Vorgang benötigt etwas Zeit für die Berechnungen. Ist alles berechnet, erklingt ein lieblicher Gong, der mitteilt, daß das Programm wieder ansprechbar ist. Beim Mitschneiden wird jedes neue Bild auf Diskette gespeichert; dadurch verzögert sich dieser Vorgang. Ein kleiner Trick: Das Frame nicht gleich mitspeichern, sondern nur die Framevorgänge, danach noch einmal laufenlassen und dabei mitspeichern (und, wenn man mag, in aller Ruhe eine Zigarette rauchen). Jetzt ist der Clip fertig, nun wollen wir ihn auch „schnell" laufen lassen.

Dazu müssen wir das Programm VideoScape 3D verlassen und uns ins CLI begeben. Dort befinden sich zwei Programme: „Showanim und Playanim". Eine Anleitung zu Showanim gibt es auf der Diskette auch. Das Textfile nennt sich „Showanim. doc" und wird genauso ausgedruckt wie das oben beschriebene Textfile. Mit Showanim kann eine Animationssequenz geladen und abgespielt werden. Hier können die Anzahl der Durchläufe und die Geschwindigkeit bestimmt werden. Die Sache hat jedoch einen Haken: Die Files werden in einem Schwung in den Speicher geladen. Wenn nun das File größer ist als der vorhandene Speicher, bricht das Programm mit einer Fehlermeldung ab. Ein 1MByte Amiga oder mehr ist gerade hier sinnvoll. Mit diesen beiden Programmen hat man die Möglichkeit, einen, wenn auch sehr speicherintensiven, Vorspann zu machen, indem der Aufruf einfach in die Startup-Sequence eingebaut wird. Die Handhabung dieser beiden Programme ist nicht schwierig und wird in dem kleinen Tutor ausführlich beschrieben, deshalb möchte hier nicht genauer darauf eingehen.

Zurück zur Workbench. Dort befinden sich noch zwei weitere ganz nützliche Tools: Das Object-Composition-Tool, kurz OCT, und der Easy-Geometry-Generator oder EGG.

OCT kann eines oder mehrere Objekte laden und sie verändern, um sie danach wieder abzuspeichern. Die Farbe und Form eines Objektes kann hier beliebig geändert werden, indem die entsprechenden Koordinaten eingegeben werden. Ein wirklich einfach zu bedienendes, aber sehr nützliches Programm.

Der EGG ist eine stark vereinfachte Form des Designers. Man hat die Wahl zwischen neuen Objektformen, von einer Box über Zylinder bis hin zu einem Bergmassiv. Abhängig von der Wahl fragt das Programm die nötigen Größen ab, bei einem Quader zum Beispiel die Kantenlänge. Das Objekt ist mit diesem Programm nicht zu besichtigen, kann aber abgespeichert und verwendet werden.

Alles in allem bietet dieses umfangreiche und durch das User-Manual gut beschriebene Softwarepaket alles, was zu einem Animationsprogramm gehört. 3D-Grafiken animieren zu lassen, und das in hoher Geschwindigkeit, ist ein wirklicher Kaufanreiz. Geschrieben wurde das Programm übrigens mit Aztec C.

Kombiniert mit Deluxe Video II, Amiga Art Machine Grafiken und dem 2D-Animator kann grafisch alles aus dem Amiga „herausgeholt" werden. Für Hobbyfilmer, die ihren Amiga zu mehr als nur zum Einblenden von Text verwenden wollen, ist dieses Programm das Richtige. Aber auch für Laien an diesem Computer ist es wegen seiner leichten Handhabung zu empfehlen. Ich möchte mich Colin French anschliessen, der die Anleitung zu Designer 3D verfasst hat: HAVE FUN!

Thomas Schlereth

Bezugsquelle:
DTM, Wiesbaden, ☎ 0 69 / 707 11 02
Softline, Oberkirch

Preis: DM 398,–



Kein Bild ist vor ihm sicher! Der BUTCHER schlägt zu!

BUTCHER, also 'Metzger', ist sicher ein ungewöhnlicher Name für ein Grafikprogramm, und auch das dazugehörige Piktogramm (Icon) bringt keine Aufklärung, denn es handelt sich dabei um ein nach unten klappendes Metzgerbeil. Was verbirgt sich hinter diesem skurrilen Namen für ein Programm?

BUTCHER ist vorwiegend ein Hilfsprogramm, mit dem die verschiedensten Bildformat untereinander konvertiert werden können. Dies bedeutet, daß es ein Bild, das mit einem bestimmten Malprogramm oder einem Digitizer erstellt wurde, in ein beliebiges Format umwandeln kann. Die einzige Voraussetzung ist, daß die Bilddatei im IFF-Format vorliegt. Da jedoch fast alle Programme dieses Standard-Format verwenden, dürften dabei keine Probleme auftreten.

BUTCHER ist kein Malprogramm. Es bietet daher nur begrenzte Möglichkeiten, Bilder zu erstellen, stattdessen aber viele Funktionen zum Bearbeiten und Manipulieren schon erstellter Bilder.

BUTCHER zeichnet sich unter anderem durch ein intelligentes Dateilademenü aus, das auch während eines Ladevorgangs noch auf die Aktionen des Benutzers reagiert. Dadurch kann man jederzeit das Laufwerk wechseln und eine Datei dann schon anwählen, wenn das Verzeichnis noch nicht komplett geladen ist. Andere Programmierer könnten sich daran ein Beispiel nehmen. Das jeweilige Bildformat wird automatisch erkannt und der Bildschirm entsprechend eingestellt.

Bäumchen wechsle dich...

Die einfachste Bildmanipulation besteht darin, die Auflösungsstufe eines Bildes zu ändern. In zwei Schritten wird eine Wandlung der horizontalen (320 < – > 640) und vertikalen (200 < – > 400) Punkte erreicht, so daß aus einem LORES- ein HIRES-Bild und umgekehrt wird. Außerdem läßt sich ein LORES-Bild in das HAM-Format umwandeln. Dieses Format kann jedoch vom BUTCHER nicht weiter verarbeitet werden und ist deshalb nur zum Einlesen bzw. abschließenden Abspeichern geeignet.

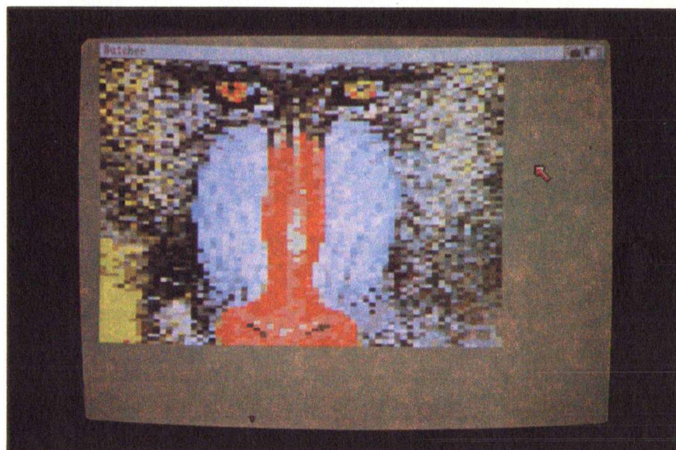
Reduzieren ist angesagt

Eine statistische Auswertung der Farbverteilung erhält man mit 'HISTOGRAMM'. Das ist jedoch noch lange nicht alles, was diese Funktion zu leisten vermag. Farbbreiche können bestimmt, sortiert und ausgetauscht werden. Außerdem kann die Anzahl der Farben reduziert werden, indem man ähnliche Farben zu einer zusammenmischt. Diese Funktion kann auch automatisch durchgeführt werden, wobei die Anzahl der Farben, in Abhängigkeit von den verwendeten Bitplanes, stufenlos eingestellt werden kann. Wenn man jetzt noch die Anzahl der Planes reduziert, dann ist der Speicherplatz des Bildes erheblich geringer. Dieser gesamte Prozeß kann auch automatisch vollzogen werden, wenn 'AUTO CHOP' angewählt wird. BUTCHER sucht dann nach den Farben, die sich am besten zusammenfassen lassen und reduziert so deren Anzahl, bis es sich mit der Zahl der Bitplanes vereinbaren läßt.

Für jeden etwas – Special Effects

EDGE ist eine Funktion, die 'Ecken' in einem Bild aufspürt und diese weiß, schwarz oder farbig hervorhebt. Leitet man diese Konturen auf ein leeres Blatt, dann zeichnet sich nur eine Umrißlinie des Objekts ab. Ein comicartiger Abzug des Originals entsteht. Der Effekt läßt sich über zwei Regler manipulieren.

FILTER beseitigt alleinstehende Farbpixels oder keinere Gruppen ei-



Eine grobe Auflösung: MOSAIC

ner bestimmten Farbe, indem es sie mit der Nachbarfarbe übermalt. Diese Funktion ist hauptsächlich dazu geeignet, eine Farbe, die nur mit kleinen 'Farbspritzern' im Bild auftritt, zu entfernen, um dieses Farbregister anderweilig zu verwenden.

MOSAIC entspricht genau dem, was man sich darunter vorstellt. Das Ursprungsbild wird in Vierecke zerlegt und damit verfremdet.

SLICE PLANE ist ebenfalls eine interessante Funktion, denn damit können die einzelnen Bitplans gezielt an- und ausgeschaltet werden, woduch sich interessante Effekte ergeben, die bei Farbbildern natürlich zu Falschfarben führen.

Die Option DENSITY SLICE erlaubt im Normalfall das schrittweise Ausblenden der jeweils dunkelsten bzw. hellsten Farbe der Palette.

Aufs Papier damit

Zum Ausdrucken verwendet BUTCHER den Druckertreiber der Workbench, was nichts besonderes ist. Schon interessanter ist die Möglichkeit, nur einen bestimmten Bildschirmbereich auszudrucken. Dieser Bereich kann auch auf die komplette Bildbreite vergrößert werden. Die Qualität des Ausdrucks hängt dabei naturgemäß stark vom verwendeten Drucker und Druckertreiber ab.



Manipulationen an der Farbverteilung

Der große 'Manipulator'

BUTCHER ist ein nützliches Programm, das es erlaubt, sämtliche Formate ineinander umzuwandeln. Außerdem ist es eine unentbehrliche Hilfe, wenn man die Anzahl der Bitplanes bzw. der Farben ändern will oder muß (um Speicherplatz zu sparen!). Viele dieser Funktionen sind 'von Hand' gar nicht durchführbar. Ein weiteres interessantes Gebiet von BUTCHER sind die 'Special Effects', also Funktionen, die ein gestalterisches und künstlerisches Bearbeiten der Bilder erlauben. BUTCHER ist, im Bereich seiner Funktionen, ein unentbehrliches Hilfsmittel und ist deshalb jedem kreativen AMIGA-Anwender sehr zu empfehlen. Der Preis von knapp hundert Mark ist in Anbetracht der Leistungen gerechtfertigt.

(mn)

Features:

- Verarbeitet und wandelt sämtliche Bildformate (IFF-Format)
- Unterstützt das Reduziern von Farben und Bitplanes
- Viele Funktionen zum gestalterischen Bearbeiten der Bilder
- Ausdruck von Bildausschnitten mit Vergrößerung

Bezugsquelle:
Intelligent Memory
Tel.: 069/7071102

Preis: ca. 89,– DM



Das KICKSTART-PUZZLE

Sag mir, wo die Teile sind

Das folgende Amiga-Basic Programm ermöglicht es, ein Bild von Diskette in beliebiger Auflösung (von Lowres bis HAM) einzuladen und in das Puzzlespiel zu übertragen, welches es kräftig schüttelt und den Benutzer mit der Aufgabe beauftragt, es wieder in den Urzustand zu bringen. Viel Spaß!

Der Ursprung

Dieses Basic-Programm entstand beim Durchforsten der Basic-Befehlsliste. Ausschlaggebend waren die Befehle PUT und GET, mit denen man Bildschirmteile in Variablen ablegen kann und von dort wieder zurück auf den Bildschirm. Also, das Prinzip ist schon geboren. Ein Bild wird eingeladen (dazu später), ein Bereich ausgewählt, dieser in viele kleine Teile zerlegt und gemeinerweise gemischt auf dem Bildschirm wieder ausgegeben. Jetzt muß man mit der Maus und natürlich mit scharfen Augen den Bildausschnitt zusammenpuzzeln. Wie schwer das ist, liegt selbstverständlich an der Anzahl der Teile. Diese ist frei variierbar und kann von 1 bis endlich gehen. Wie weit endlich nun geht, liegt an Ihren Adleraugen, mehr als 10★10, also 100 Teile, sind allerdings schon fast unpuzzlebar. Aber probieren Sie es doch einfach aus.

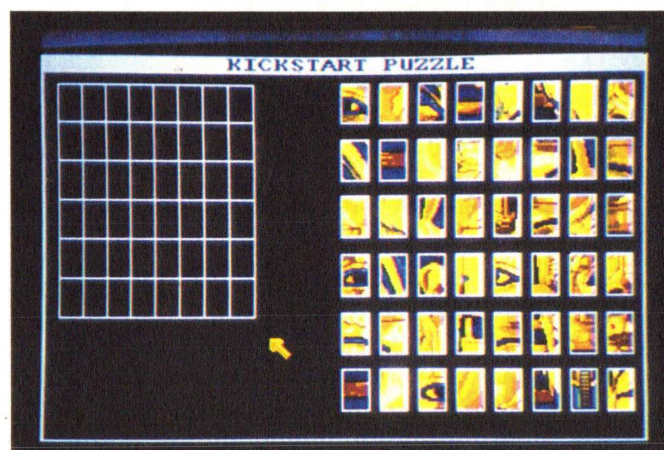
Die Sache mit dem Klick

Angewählt wird ein Teil, wie auch sonst, per Mausklick. Anschließend bewegt man die Maus auf die Stelle in der linken Bildschirmhälfte, von der man glaubt, daß es die richtige ist. Per Klick wird das Teil dort plaziert. An dieser Stelle muß erwähnt werden, daß das erste Anklicken nach Programmstart etwas länger dauert. Beim zweiten Teil erfolgt das Anwählen sofort. Der Grund: Eine kleine Macke das Amiga-Basic, das beim Erstdurchlaufen eines Programmabschnitts dieses erst kennenlernen muß. Also ein wenig Geduld (circa drei Sekunden), das Amiga-Basic ist eben bei neuen Programmen etwas scheu – eine durchaus menschliche Eigenschaft.

Jetzt geht es wesentlich schneller: Anklicken – rüberschieben – klick, das Teil ist plaziert. Ist ein Teil an der falschen Stelle, was natürlich trotz der besten Augen mal passieren kann, so klickt man es einfach an und schiebt es an eine andere Stelle. Will man ein Teil überhaupt nicht mehr auf der linken Seite sehen, so klickt man es an und drückt danach die DEL-Taste. Das Teil wird wieder auf der rechten Seite plaziert.

Der Bildlader

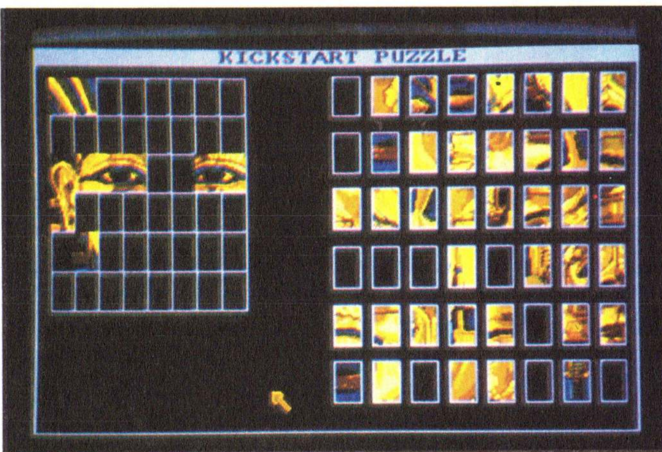
Ein beliebiges Bild im ACBM-Format wird vom Unterprogramm 'Main' auf den Bildschirm geladen. Das ACBM-Format wurde deshalb gewählt, weil es von Basic aus schnell zu laden ist. Zwar ist das IFF-Format häufiger anzutreffen, doch ist die Ladezeit eines solchen Formats zu nervenquälend, und lange Wartezeiten kann kein Programmierer vertragen. Wenn man aber gerne ein IFF-Bild puzzeln würde, so bedient man sich einfach des Formatwandlers 'LOADILBM-SAVEACBM' von der Original-Basicdiskette 'EXTRAD'. Die Laderoutine in diesem Programm wurde ebenfalls von dieser Diskette entnommen. Das bietet den Vorteil, daß Sie den Programmteil ab dem Label MAIN: nicht abtippen müssen, sondern einfach hinzumergen können. Dazu mit dem Cursor ans Ende des Programms und nun im Direktmodus eintippen:
MERGE "LoadACBM"



Zerlegt in 48 Teile ist wenig zu erkennen

Die einzige Änderung, die vorzunehmen ist, betrifft den Window-Aufruf in Zeile 422. Dieser ist gemäß Listing abzuändern, da sonst der Windowrandbereich nicht genutzt werden kann. Weiterhin können Sie einige Routinen nach dem Mergen entfernen.

Die Laderoutine sieht zugegeben etwas kompliziert aus, aber dafür ist sie flexibel. Sie erkennt nämlich die Auflösung sowie die Anzahl der Bitplanes und generiert automatisch einen SCREEN der betreffenden Konstellation.



Die ersten Teile lassen ahnen...

Die einzelnen Routinen

Unser Programm wählt nun aus diesem Bild einen Ausschnitt aus, dies erledigt das Unterprogramm AUSSCHNITT. Es erscheint ein flackerndes Rechteck, das der Maus folgt. Per Mausklick wird nun dieser Ausschnitt ins Puzzle übernommen.

Die Routine AUSLESEN unterteilt nun diesen Ausschnitt in die angegebene Menge Teile (nx,ny). Die Teile werden im Feld P%(0,nx,ny) abgelegt. Die Dimensionierung erfolgt am Programmanfang und ist nicht gerade einfach. Das Anleitungsbuch schweigt sich jedenfalls dazu aus. Das Prinzip ist folgendes: Ein Bildschirmausschnitt wird in einer Integer-Variable abgelegt. Da eine solche Variable nur zwei Bytes lang ist, muß man sie auf die benötigte Größe dimensionieren, also als Feld tarnen.

Diese Methode reichlich seltsam, aber die Väter des Amiga-Basic fanden es wohl gut. Einerseits muß man die Größe vorher berechnen und zwar nicht zu klein, da sonst unser allseits bekannter GURU zur Meditation aufruft. Zweitens ist es etwas verwirrend, wenn man eine normale Variable angibt, sie jedoch als Feld dimensioniert. Doch sei es wie es ist, man muß damit leben.

Aus zwei macht drei

Nach einigen Meditationsrunden wurde der Dreh klar. Will man ein zweidimensionales Feld kreieren, um die Daten der einzelnen Puzzleteile aufzunehmen, so dimensioniert man es mit

DIM P%(Laenge,nx,ny)

dreidimensional. Die Variable 'Laenge' enthält die Speicherplatzmenge eines Puzzleteiles. Nx und Ny die Anzahl der Teile in X- und Y-Richtung. Ein solches Teil spricht man dann z. B. mit

GET (x1,y1) – (x2,y2),P%(0,Nx,Ny)

an. Die Null sollte unbedingt beibehalten werden, da sie angibt, daß der Block mit diesem Element beginnt. Ein solcher Block ist folgendermaßen aufgebaut:

Wort 1: Breite des Blocks in Pixel
Wort 2: Höhe des Blocks in Pixel
Wort 3: Anzahl der Bit-Ebenen
Wort 4: Laenge: Bildinformation

Wort1 liegt in Zelle P%(0,Nx,Ny), die Breite in Zelle P%(1,Nx,Ny), usw. . Gibt man aus Versehen bei PUT das zweite Element an, so kommt der Amiga völlig durcheinander, da er nun die Höhe für die Breite hält, die Anzahl der Bit-Ebenen für die Höhe – Guru erscheint prompt.

Die beim Dimensionieren anzugebende Länge wird in der selbstdefinierten Funktion MEM berechnet.

DEF FNmem(x,y) = 6 + (y + 1) ★ INT ((x + 16)/16) ★ Planes

Der Hintergrund: In X-Richtung muß die Breite immer auf die nächste Wortgröße aufgerundet werden. Multipliziert mit der Höhe + 1 und addiert zu 6 (6 Bytes = 3 Wörter für obige Information) ergibt das den Platzbedarf eines Puzzleteils. 'Höhe + 1' weil z. B. ein Block von y1 = 10 bis y2 = 20 zwar 10 hoch ist (20 – 10), aber genaugenommen doch aus 11 Linien besteht.

Mischmasch

Als nächstes kommt das Mischen. Die Routine schüttelt die Teile kräftig durcheinander, da sonst das Puzzle zu einfach wäre. Die Routine 'Ausgabe' kümmert sich nun um die Ausgabe der gemischten Teile auf der rechten Bildschirmhälfte, wonach 'Raster' ein Raster zeichnet, um die Position der Teile anzudeuten. Bei manchen Bildern kommt es leider vor, daß die Rahmenfarbe gleich dem Hintergrund ist. Das liegt an dem jeweiligen Bild und kann nicht vorausgesehen werden. Zur Not kann man ja dann in der Routine Raster die Linienfarbe ändern (z. B. COLOR 3).

Kontrolle

Das Hauptprogramm besteht nun darin, die Maus und die Tastatur zu überwachen. Wurde die HELP-Taste gedrückt, so geschieht etwas erfreuliches. Der Rechner zeigt Ihnen dann nämlich bis zum nächsten Tastendruck das Bild so, wie es auszusehen hat, an. Eine sehr wichtige Funktion, da man sonst leicht verzweifelt.

Wurde die Maustaste gedrückt (Mouse(0) < 0), so verzweigt das Programm in Abhängigkeit von der Maus-X-Position in die Programmteile zum Anwählen und Positionieren oder zur Routine Ändern. Im ersten Fall wird ein neues Teil aus der rechten Bildschirmhälfte angewählt und links positioniert, im anderen Fall ein Teil links angewählt und auf ein anderes linkes Feld transportiert.

Die gezähmte Maus

Das war eigentlich schon die Hauptschleife. Bleiben noch einige Routinen, die in Folge noch kurz erläutert werden. Zunächst 'Mausr' und 'Mausl'. Sie wandeln die absoluten Mauskoordinaten in Werte von '0 bis Nx – 1' bzw. '0 bis Ny – 1'. Damit

können die Felder direkt angesprochen werden.

Sag mir, wo die Teile sind

Die Routine 'Löschen' sorgt dafür, daß ein angewähltes Teil von der linken Bildschirmhälfte gelöscht wird. Dies geschieht durch Anklicken des Teiles und anschließendes Drücken der DEL-Taste. Die letzte Routine 'Neuaufbau' wird aufgerufen, sobald ein Puzzleteil aus der linken Bildschirmhälfte gelöscht oder verschoben wurde. Sodann zeichnet sie das Raster neu und plaziert, falls vorhanden, die Teile neu darüber. Bei größeren Puzzles hält dies den Rechner ein wenig auf, aber das ist halt Amiga-Basic. Es wird Zeit, daß Firmen wie OMIKRON ihr Basic vom ST auf den Amiga umsetzen.

Ein Wort zum Speicher

Das Programm verkraftet alle verschiedenen ACBM-Bildformate mit endlich vielen Puzzleteilen. Wie endlich ist einerseits von der jeweiligen Auflösung und von dem verfügbaren Speicherplatz abhängig, da der Bedarf mit steigender Teilezahl ansteigt. Hier müssen wir auch einen Wermutstropfen für die Besitzer eines Amiga 500 oder Amiga 1000 ohne Megabyte-Speichererweiterung ausschenken. Bilder mit hoher Auflösung (z. B. Interlace), womöglich noch mit vielen Farben, überfordern den 'Kleinen' leider hoffnungslos.

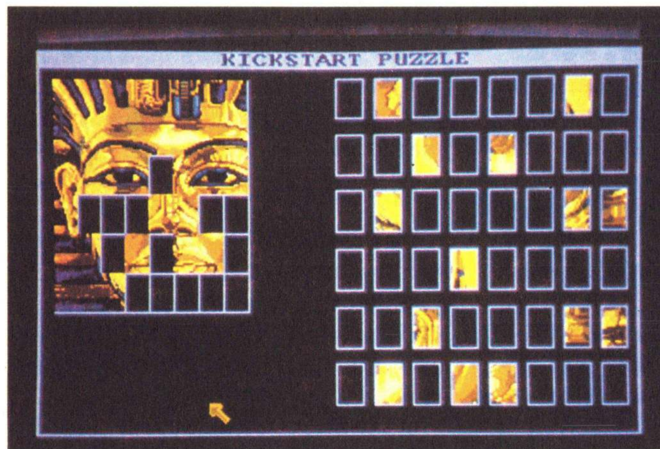
Anders mit Bildern niedriger Auflösung. Diese verkraftet der Amiga 500 problemlos, wobei man allerdings keine andere Task laufen lassen darf und das Basic nicht von der Workbench, sondern von CLI aus startet (load wb weglassen). Am besten ist, wenn die Startupsequenz leer ist, d. h. garnichts reinschreibt. Dazu generiert man mit einem Editor (ED) einen Text ohne Inhalt und speichert ihn unter dem Namen 'startup-sequence' auf die Bootdiskette ab. So gewinnt man noch ein wenig Speicher dazu, oder besser gesagt: man verliert weniger. Nach Beachtung dieser Punkte kann man sogar Bilder mittlerer Auflösung laden (640★200). Da dem Basic normalerweise lediglich 25.000 Bytes an internem Speicher zur Verfügung stehen und das auch für Bilder kleiner Auflösung zu wenig wäre, müssen wir uns ein wenig mehr reservieren. Dazu dient der CLEAR-Befehl. Ihn gibt man am besten im Direktmodus ein, vor dem Starten des Programms, da er sich im Programm meist weigert, zum zweiten Mal Speicher bereitzustellen. Diese große Speichermenge wird benötigt zum Dimensionieren der Felder, besonders derer, die Bildausschnitte enthalten. Nach folgender Liste zeigten sich keine Probleme:

```
CLEAR ,60000& ' low-res
CLEAR ,100000& ' med-res
CLEAR ,200000& ' hi-res
```

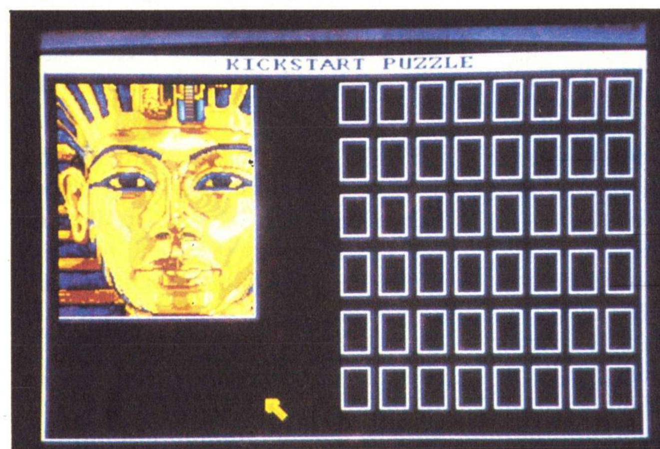
Falls sich während des Programmablaufs eine Fehlermeldung ergibt 'Out of Memory', so erhöhen Sie den Wert einfach. Haben Sie einen Mega-Rechner, dann nehmen Sie sich einfach 200 KByte und kümmern sich um nichts mehr. Aber wenn's nicht unbedingt Interlace sein muß, läuft das Programm auf dem 500er ohne Probleme.

Nun aber viel Spaß beim Abtippen. Die Programmlänge hält sich einigermaßen in Grenzen, besonders wenn man den Ladeteil übernimmt. Natürlich kann man auch einen IFF-Lader einbauen oder beides, das liegt im persönlichen Interesse.

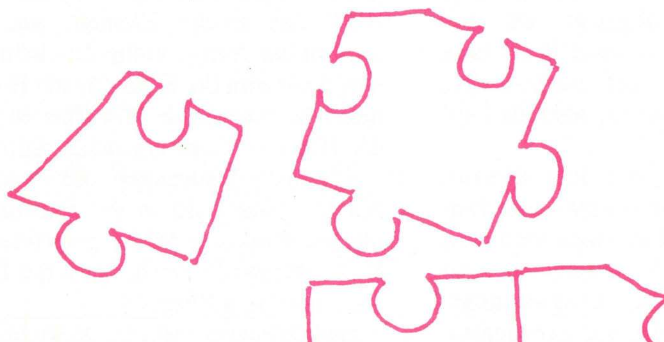
(HS)



...um was es sich handelt



Geschafft!



```
'****************************************
'*                                      *
'*                                      *
'*  K I C K S T A R T -  P U Z Z L E    *
'*                                      *
'*   with a KICK                        *
'*                                      *
'*   by Harald            8/87          *
'*                                      *
'*                                      *
'* in AMIGA Basic                       *
'*                                      *
'* Viel Spass  beim Puzzeln             *
'*                                      *
'****************************************


' Einer der folgenden CLEAR-Befehle muß im Direktmodus
' gegeben werden
'CLEAR ,60000&      ' low-res
'CLEAR ,100000&     ' med-res
'CLEAR ,200000&     ' Interlace will etwas mehr
PRINT "Willkommen zum KICKSTART PUZZLE"
INPUT "Wieviel Puzzleteile möchten Sie (x,y)";Nx,Ny
IF Nx>9 OR Ny>9 THEN PRINT "Wenn das gutgeht "

GOSUB main
Xmax=scrwidth% :Ymax=scrheight%
Planes=idepth% :Modus=kk

Xbild=Xmax/3 :Ybild=Ymax*.6   'Größe des Ausschnitts

Xbild=(Xbild \ Nx)*Nx  ' Runden auf Zahl, die teil-
Ybild=(Ybild \ Ny)*Ny  ' bar ist durch nx bzw. ny
Xmax=(Xmax \ Nx)*Nx
Ymax=(Ymax\ Ny)*Ny

DEF FNmem(X,Y)=6+(Y+1)*INT((X+16)/16)*Planes  'Platz in Bytes/2

Gx=Xbild/Nx     ' Breite eines Puzzleteils
Gy=Ybild/Ny     ' Höhe eines Puzzleteils

Mixx=Xmax/2     ' Bildschirmstelle, die beim Mischen
Mixy=Ybild/2    ' mißbraucht wid

DIM Z(Nx*Ny)    ' Feld zum Mischen
DIM Qx(Nx,Ny)   ' X-Position der einzelnen Teile  (rechts)
DIM Qy(Nx,Ny)   ' Y-    "     "        "      "   (links)
DIM Mr(Nx,Ny)   ' Merkt ob das betreffende Teil noch vorhanden ist

DIM Ml(Nx,Ny)   ' Merkt nach anwählen die alte Position des Teils

DIM Mm(Nx,Ny)   ' Merkt ob betr. Platz im linken Feld noch frei is
t
DIM A%(FNmem(Xbild,Ybild))     ' Kompletter Ausschnitt
DIM Hilf%(FNmem(Xbild,Ybild)) ' Enthält momentanes Puzzle (links)

                               ' beim Anzeigen
Blocklaenge=FNmem(Gx,Gy)       ' So viel Platz braucht ein Teil

DIM P%(Blocklaenge,Nx,Ny)      ' Enthält das Teilefeld

DIM Select%(Blocklaenge)       ' Enthält das angewählte Teil

GOSUB Ausschnitt    '  gibt xeck und yeck

  GET (Xeck,Yeck)-(Xeck+Xbild-1,Yeck+Ybild-1),A%  'Hole Ausschnit
  t
  WINDOW CLOSE 2
  WINDOW 2,"               KICKSTART PUZZLE ",,0,2

  CLS
  PUT (5,5),A%,PSET   ' und zeige ihn an

  GOSUB Mischen        ' sonst wäre es ja zu leicht
  GOSUB Auslesen       ' Lies die Einzelteile aus dem Bild
  GOSUB Ausgabe        ' zeige die Teile rechts an
  LINE (5,5)-(5+Xbild-1,5+Ybild-1),0,bf  ' weg damit

  GOSUB Raster         ' Malt ein Raster

  Label1:               ' Repeat Until gibt's ja leider nicht

    Dum=MOUSE(0)        ' Wecke die Maus aus ihrem Schlaf
    In$=INKEY$          ' Tastendruck erfragen
    IF In$ <> "" THEN   ' Wurde eine Taste gedrueckt ?

      IF ASC(In$)=139 THEN ' War es auch noch die HELP-Taste

        GET (5,5)-(5+Xbild-1,5+Ybild-1),Hilf%  ' sichere momen-
                                                ' tanes Bild
        PUT (5,5),A%,PSET ' zeige die Lösung
        WHILE INKEY$=""   ' und warte bis Taste gedrueckt wird
        WEND
        PUT (5,5),Hilf%,PSET  ' zeige wieder momentanes Bild an

      END IF
     END IF

   IF MOUSE(0)<0 AND MOUSE(1)>Xmax/2  THEN 'Maus gedrueckt und in
    rechter Bildhälfte
      GOSUB Anwaehlen       ' Wähle ein Teil aus (rechts)
      GOSUB Positionieren  ' und positioniere es (links)
      WHILE MOUSE(0)<0 :WEND
    END IF

    IF MOUSE(0)<0 AND MOUSE(1)<Xbild+5 THEN

      GOSUB Aendern
      WHILE MOUSE(0)<0 : WEND
  END IF

GOTO Label1

END
'
Anwaehlen:
   Flag1=0
   GOSUB mausr
   X1=Qx(Mausx,Mausy)
   Y1=Qy(Mausx,Mausy)
   X2=X1+Gx
   Y2=Y1+Gy
   IF Mr(Mausx,Mausy)=0 THEN
     Mr(Mausx,Mausy)=1
     Flag1=1
     GET (X1,Y1)-(X2-1,Y2-1),Select%
     LINE (X1,Y1)-(X2-1,Y2-1),0,bf
     Ml=Mausx+Mausy*Nx
   END IF
RETURN

Aendern:
'
  Flag1=0
  GOSUB mausl
  IF Mm(Mausx,Mausy)=1 THEN ' teil vorhanden ?
    Flag1=1   ' angewählt
    X1=5+Mausx*Gx      : X2=X1+Gx-1
    Y1=5+Mausy*Gy      : Y2=Y1+Gy-1
    Ml=Ml(Mausx,Mausy)
    Mm(Mausx,Mausy)=0  ' nix mehr drin
    GET (X1,Y1)-(X2,Y2),Select%  ' Teil holen
    LINE (X1,Y1)-(X2,Y2),0,bf    ' alten Platz löschen
    WHILE MOUSE(0)<0
    WEND

    GOSUB Positionieren
    GOSUB neuaufbau
  END IF
RETURN


RETURN

'
Auslesen:

  FOR X=0 TO Nx-1
    FOR Y=0 TO Ny-1
      We=X+Y*Nx
      GET (5+Gx*X,5+Gy*Y)-(5+Gx*(X+1)-1,5+Gy*(Y+1)-1),P%(0,(Z(We)
       MOD Nx),(Z(We)\Nx))
      ' Ein Teil aus dem Gesamtbild herauslesen
    NEXT Y
  NEXT X
RETURN
'
'
Ausgabe:
  FOR X=0 TO Nx-1
    FOR Y=0 TO Ny-1
      Qx(X,Y)=INT(Xmax/2+Gx*X+X*(Xmax/2-Xbild)/Nx)-5
      'X-Position berechnen
      Qy(X,Y)=INT(5+Y*Gy+Y*(Ymax-Ybild-Gy)/Ny)
      'Y-Position berechnen
      PUT (Qx(X,Y),Qy(X,Y)),P%(0,X,Y),PSET
      'dorthin ausgeben
      LINE (Qx(X,Y)-1,Qy(X,Y)-1)-(Qx(X,Y)+Gx,Qy(X,Y)+Gy),1,b
      'Rahmen darum ziehen
    NEXT Y
  NEXT X
RETURN
'

mausr:
  Dum=MOUSE(0)
' Rechnet die Mauskoordinaten um auf die Zahlen 0..nx-1 ,bzw. 0..
ny-1
  Mausx=INT((MOUSE(3)-Xmax/2)/Xmax*2*Nx )
  Mausy=INT(MOUSE(4)/(Ymax-Gy)*Ny)
  IF Mausy>=Ny OR Mausy<0 THEN Dum=MOUSE(0):GOTO mausr
  'Falls außerhalb: nochmal
  IF Mausx>=Nx OR Mausx<0 THEN Dum=MOUSE(0):GOTO mausr
RETURN
'
'
mausl:
 'wie oben, aber fuer linkes Feld
  Dum=MOUSE(0)
  Mausx=INT((MOUSE(3)-5)/Xbild*Nx)
  Mausy=INT((MOUSE(4)-5)/Ybild*Ny)
  IF Mausy>=Ny OR Mausy<0 THEN Dum=MOUSE(0):GOTO mausl
  IF Mausx>=Nx OR Mausx<0 THEN Dum=MOUSE(0):GOTO mausl
RETURN
'
'
'
Raster:
' Legt ueber das Bild (links) ein Raster
  COLOR 1
  FOR X=0 TO Nx
    LINE (5+Gx*X,5)-(5+Gx*X,5+Ybild)  '
  NEXT X
  FOR Y=0 TO Ny
    LINE (5,5+Gy*Y)-(5+Xbild,5+Gy*Y)
  NEXT Y
RETURN
'
'
'
Positionieren:

  Flag2=0
  IF Flag1=1 THEN

   WHILE -1
     Dum=MOUSE(0)
     In$=INKEY$
      IF In$<>"" THEN
       IF ASC(In$)=127 THEN
         GOSUB Loeschen
         GOTO Lab3
       END IF
     ELSE
       IF MOUSE(0)<0 AND MOUSE(1)<Xbild +5 THEN
         GOSUB mausl
         IF Mm(Mausx,Mausy)=0 THEN
           Mm(Mausx,Mausy)=1
           Ml(Mausx,Mausy)=Ml
```

```
                 PUT (5+Gx*Mausx,5+Gy*Mausy),Select%,PSET
                 GOTO Lab3
               END IF
             END IF
           END IF
         WEND
         Lab3:
          Flag1=0
       END IF
     RETURN
     '
     '
     Loeschen:
       GOSUB mausl

       LINE (5+Gx*Mausx,5+Gy*Mausy)-(5+Gx*(Mausx+1),5+Gy*(Mausy+1)),0,
       bf

     · Ky=INT(Ml(Mausx,Mausy) \ Nx)
       Kx=INT(Ml(Mausx,Mausy) MOD Nx)
       PUT (Qx(Kx,Ky),Qy(Kx,Ky)),P%(0,Kx,Ky),PSET
       Mr(Kx,Ky)=0
       Mm(Mausx,Mausy)=0
       REM GOSUB neuaufbau
     RETURN
     '
     '
     neuaufbau:
       GOSUB Raster
       FOR X=0 TO Nx-1
         FOR Y=0 TO Ny-1
           IF Mm(X,Y)=1 THEN
             Ky=INT(Ml(X,Y) \ Nx)
             Kx=INT(Ml(X,Y) MOD Nx)
             PUT (5+Gx*X,5+Gy*Y),P%(0,Kx,Ky),PSET
           END IF
         NEXT Y
       NEXT X
     RETURN
     '
     '
     '
     Ausschnitt:
       '  Zeichnet eine inbertieremde Box zum Ausschnitt wählen

       WHILE MOUSE(0)>-1  OR MOUSE(1)>Xmax-Xbild OR MOUSE(2)>Ymax-Ybil
       d
         X=MOUSE(1):Y=MOUSE(2)
         GET (X,Y)-(X+Xbild-1,Y+Ybild-1),A%
         PUT (X,Y),A%,XOR
         PUT (X,Y),A%,XOR
       WEND
       Xeck=MOUSE(3):Yeck=MOUSE(4)
     RETURN
     '
     Mischen:

      RANDOMIZE TIMER  ' Zufallsgenerator initialisieren

        FOR I=0 TO Nx*Ny-1 : Z(I)=I :NEXT I
          FOR I=0 TO Nx*Ny-1
          zuf=INT(RND*Nx*Ny)
          SWAP Z(I),Z(zuf)
        NEXT I
     RETURN
     '

     ' ************************************************************


     ' Der folgende Programmteil kann von der EXTRAD-Diskette
     ' hinzugemergt werden   (siehe Anleitung)

     main:

     DIM bPlane&(5), cTabWork%(32), cTabSave%(32)

     DECLARE FUNCTION xOpen&  LIBRARY
     DECLARE FUNCTION xRead&  LIBRARY
     DECLARE FUNCTION xWrite& LIBRARY
     DECLARE FUNCTION AllocMem&() LIBRARY
     PRINT:PRINT "Suchen nach .bmap-Dateien ... ";
     LIBRARY "dos.library"
     LIBRARY "exec.library"
     LIBRARY "graphics.library"
     PRINT "Libraries gefunden"

     PRINT "ACBM-Datei-Namen eingeben (ggf. incl. Zugriffspfad):"
     INPUT "   ACBM-Dateiname = ";ACBMname$
     REM - ACBM-Bild laden
     loadError$ = ""
     GOSUB LoadACBM
     IF loadError$ <> "" THEN GOTO Mcleanup

     RETURN




     Mcleanup:
     FOR de = 1 TO 20000:NEXT
     WINDOW CLOSE 2
     SCREEN CLOSE 2
     LIBRARY CLOSE
     IF loadError$ <> "" THEN PRINT loadError$
     END



     LoadACBM:
        REM - Variablen initialisieren
        F$ = ACBMname$
        fHandle& = 0
        mybuf& = 0
        foundBMHD = 0
        foundCMAP = 0
        foundCAMG = 0
        foundCCRT = 0
        foundABIT = 0

        filename$ = F$ + CHR$(0)
        fHandle& = xOpen&(SADD(filename$),1005)
        IF fHandle& = 0 THEN
           loadError$ = "Eingabedatei nicht gefunden/lesbar."
           GOTO Lcleanup
        END IF


        REM - Pufferspeicherplatz reservieren
        ClearPublic& = 65537&
        mybufsize& = 360
        mybuf& = AllocMem&(mybufsize&,ClearPublic&)
        IF mybuf& = 0 THEN
           loadError$ = "Pufferspeicherplatz nicht verfuegbar."
           GOTO Lcleanup
        END IF

        inbuf& = mybuf&
        cbuf& = mybuf& + 120
        ctab& = mybuf& + 240


        REM - Eingabe sollte lauten  FORMnnnnACBM
        rLen& = xRead&(fHandle&,inbuf&,12)
        tt$ = ""
        FOR kk = 8 TO 11
           tt% = PEEK(inbuf& + kk)
           tt$ = tt$ + CHR$(tt%)
        NEXT

        IF tt$ <> "ACBM" THEN
           loadError$ = "Keine ACBM-Grafikdatei."
           GOTO Lcleanup
        END IF

        REM - ACBM-Datei Chunk-weise lesen

        ChunkLoop:
        REM - Chunk-Name/Länge ermitteln
        rLen& = xRead&(fHandle&,inbuf&,8)
        icLen& = PEEKL(inbuf& + 4)
        tt$ = ""
        FOR kk = 0 TO 3
           tt% = PEEK(inbuf& + kk)
           tt$ = tt$ + CHR$(tt%)
        NEXT

        IF tt$ = "BMHD" THEN  'BitMap-Header
           foundBMHD = 1
        rLen& = xRead&(fHandle&,inbuf&,icLen&)
        iWidth%  = PEEKW(inbuf&)
        iHeight% = PEEKW(inbuf& + 2)
        idepth%  = PEEK(inbuf& + 8)
        iCompr%  = PEEK(inbuf& + 10)
        scrwidth%  = PEEKW(inbuf& + 16)
        scrheight% = PEEKW(inbuf& + 18)

        iRowBytes% = iWidth% /8
        scrRowBytes% = scrwidth% / 8
        nColors%  = 2^(idepth%)

        '" - Genug Platz fuer Videospeicher ?
        AvailRam& = FRE(-1)
        NeededRam& = ((scrwidth%/8)*scrheight%*(idepth%+1))+5000
        IF AvailRam& < NeededRam& THEN
           loadError$ = "Speicherplatz reicht nicht aus."
           GOTO Lcleanup
        END IF

        kk = 1
        IF scrwidth% >.320 THEN kk = kk + 1
        IF scrheight% > 200  THEN kk = kk + 2
        SCREEN 2,scrwidth%,scrheight%,idepth%,kk
        WINDOW 2,ACBMname$,,0,2        ' <--------- Änderung

        REM - Adressen von Screen-Structures ermitteln
        GOSUB GetScrAddrs

        REM - Schirm während Ladevorgang dunkel
        CALL LoadRGB4&(sViewPort&,ctab&,nColors%)


     ELSEIF tt$ = "CMAP" THEN  'Farbpalette
        foundCMAP = 1
        rLen& = xRead&(fHandle&,cbuf&,icLen&)

        REM - Farbpalette aufbauen
        FOR kk = 0 TO nColors% - 1
           red% = PEEK(cbuf&+(kk*3))
           gre% = PEEK(cbuf&+(kk*3)+1)
           blu% = PEEK(cbuf&+(kk*3)+2)
           regTemp% = (red%*16)+(gre%)+(blu%/16)
           POKEW(ctab&+(2*kk)),regTemp%
        NEXT


     ELSEIF tt$ = "CAMG" THEN 'Amiga ViewPort Modes
        foundCAMG = 1
        rLen& = xRead&(fHandle&,inbuf&,icLen&)
        camgModes& = PEEKL(inbuf&)


     ELSEIF tt$ = "CCRT" THEN 'Graphicraft-Farbzyklus-Daten
        foundCCRT = 1
        rLen& = xRead&(fHandle&,inbuf&,icLen&)
        ccrtDir%    = PEEKW(inbuf&)
        ccrtStart%  = PEEK(inbuf& + 2)
        ccrtEnd%    = PEEK(inbuf& + 3)
        ccrtSecs&   = PEEKL(inbuf& + 4)
        ccrtMics&   = PEEKL(inbuf& + 8)


     ELSEIF tt$ = "ABIT" THEN  'Contiguous BitMap
        foundABIT = 1
        plSize& = (scrwidth%/8) * scrheight%
        FOR pp = 0 TO idepth% -1
           rLen& = xRead&(fHandle&,bPlane&(pp),plSize&)
        NEXT

```

```
   ELSE
      REM - unbekannten Chunk-Typ lesen
      FOR kk = 1 TO icLen&
         rLen& = xRead&(fHandle&,inbuf&,1)
      NEXT
      '" - Wenn Länge ungerade, noch 1 Byte lesen
      IF (icLen& OR 1) = icLen& THEN
         rLen& = xRead&(fHandle&,inbuf&,1)
      END IF

   END IF


      REM - Fertig, wenn alle Chunks gelesen
      IF foundBMHD AND foundCMAP AND foundABIT THEN
         GOTO GoodLoad
      END IF

      REM - Lesen ok, nächsten Chunk lesen
      IF rLen& > 0 THEN GOTO ChunkLoop

      IF rLen& < 0 THEN  ' Lesefehler
         loadError$ = "Lesefehler."
         GOTO Lcleanup
      END IF

      REM - rLen& = 0  heißt EOF (Dateiende)
      IF (foundBMHD=0) OR (foundABIT=0) OR (foundCMAP=0) THEN
         loadError$ = "Wichtige IFF-Chunks nicht gefunden."
         GOTO Lcleanup
      END IF


   GoodLoad:
      loadError$ =""

      REM  Farbpalette
      IF foundCMAP THEN
         CALL LoadRGB4&(sViewPort&,ctab&,nColors%)
      END IF

   Lcleanup:
      IF fHandle& <> 0 THEN CALL xClose&(fHandle&)
      IF mybuf& <> 0 THEN CALL FreeMem&(mybuf&,mybufsize&)
   RETURN


   GetScrAddrs:
      REM - Adressen von Screen-Structures ermitteln
      sWindow&    = WINDOW(7)
      sScreen&    = PEEKL(sWindow& + 46)
      sViewPort& = sScreen& + 44
      sRastPort& = sScreen& + 84
      sColorMap& = PEEKL(sViewPort& + 4)
      colorTab&  = PEEKL(sColorMap& + 4)
      sBitMap&    = PEEKL(sRastPort& + 4)

      REM - Screen-Parameter ermitteln
      scrwidth%  = PEEKW(sScreen& + 12)
      scrheight% = PEEKW(sScreen& + 14)
      scrDepth%  = PEEK(sBitMap& + 5)
      nColors%   = 2^scrDepth%

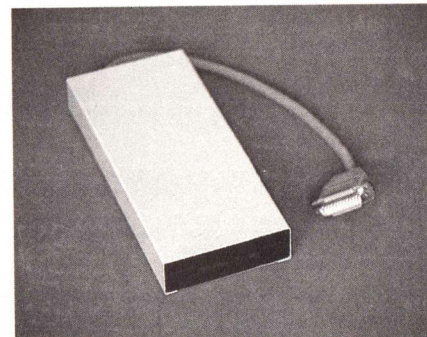
      REM - Adressen der BitPlanes ermitteln
      FOR kk = 0 TO scrDepth% - 1
         bPlane&(kk) = PEEKL(sBitMap&+8+(kk*4))
      NEXT
   RETURN
```

3.5" Laufwerk auch am PC

Die Firma PDC in Bad Homburg vertreibt schon seit längerer Zeit ein 3.5" Floppystation. Das flache Gehäuse will Farblich nicht ganz zu den verschiedenen Amigas passen. Auch das Kabel des Laufwerks ist recht kurz geraten, sodaß es beim Amiga 500 nicht neben dem Gehäuse plaziert werden kann, da es dort das Wechseln der Disketten vom internen Laufwerk stark behindert. Für die linke Seite ist das Laufwerk völlig ungeeignet, da das Kabel bis dort hin gar nicht reicht. Der einzige Platz ist der auf dem Gerät. Glücklicherweise ist auf dem Amiga 500 genug Freiraum vorhanden. Der Amiga 1000 und 2000 besitzen mit dem Laufwerk keine Platzprobleme, weil es direkt neben dem Monitor plaziert werden kann. Das Laufwerk verrichtete problemlos seine dienste und arbeitet auch auf PC Basis, d.h., daß es auch im MSDOS Format beschrieben werden kann. Dadurch ist es möglich auch MSDOS-Software auf 3.5" Disketten zu speichern. Nicht alle erhältlichen Laufwerke lassen sich mit der Sidecar oder dem Bridgeboard betreiben. Der Preis des Floppy liegt bei 369,– DM und ist unter folgender Adresse zu beziehen:

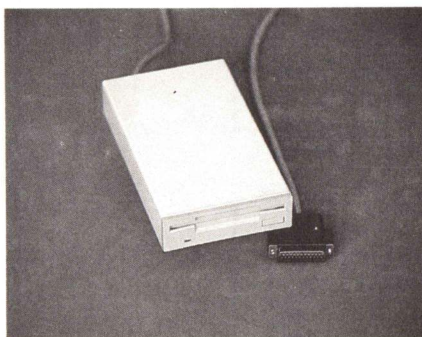
PDC
Louisenstr. 115
6380 Bad Homburg

(AK)



Neues 3.5" Laufwerk

Die neue Firma Future Vision bietet eine weitere Zusatzfloppy für den Amigo an. Das sehr kleine Gerät ist in den Farben des Amigas gehalten, sodaß es optisch hervorragend zum Gehäuse paßt. Im Innern befindet sich ein Super Slimline NEC 1036 A Laufwerk. Auf der Rückseite der Floppysration befindet sich ein EIN/AUS Schalter, der es ermöglicht das Gerät ohne entfernen des Kabels zu desaktivieren um so vielleicht nötigen Speicher zu erhalten. Einige Programme benötigen ja die 512 KByte zum reibungslosen Ablauf. Ebenfalls ist das Kabel des Laufwerks erfreulich lang, sodaß es auf dem Schreibtisch beliebig plaziert werden kann. Einziger Nachteil besteht darin, daß die Floppy nicht mit dem Bridgeboard oder der Sidecar zusammen arbeitet. Auf Wunsch, so versicherte mir der Geschäftsführer M. Stutz, wird das Gerät auch mit durchgeschleiften Bus ausgeliefert. Der Preis liegt bei 329,– DM und ist zu beziehen unter folgender Adresse:



M. Stutz
Fr.-Veith-Straße 21
6129 Höchst i. O.

(AK)

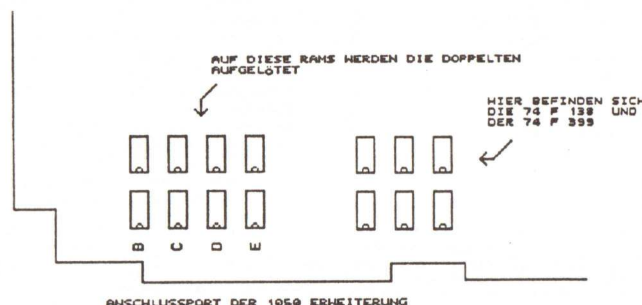
Nachtrag

In dem Artikel „Ein MEGABYTE im AMIGA 1000" haben sich leider zwei Fehler eingeschlichen. Die Leser, die auf diese Weise zu einem nicht funktionierendem Rechner kamen, möchte ich hier um Entschuldigung bitten.
Im einzelnen sind folgende Punkte zu ändern bzw. zu beachten:

1. Einige Leser hatten Schwierigkeiten mit der Lage der Bausteine. Die abgedruckte Zeichnung sollte Ihnen an diesem Punkt weiter helfen.
2. Die Pins 11 und 12 des 74F399 müssen vor dem Anlöten der Drähte von der Hauptplatine getrennt werden. Sie haben jetzt 4 (vier) Verbindungen zwischen den ICs und der Hauptplatine getrennt. Je eine an den 74F138 (Pin 3) und zwei am 74F399 (Pin 11 und 12).
3. Die Angabe zum ADDMEM Befehl muß wie folgt lauten:
   1> ADDMEM 100000 17ffff

Die Speichererweiterung läßt sich abschalten, sollte es bei einigen Programmen Schwierigkeiten geben. Dazu wird ein einfacher Schalter in die Leitung von Pin 10 des 74F399 nach Pin 3 des 74F138 gelegt. Bei Schalter geöffnet ist die Erweiterung inaktiv.

(GC)



SONDERAKTION

Preissenkung bei der MTR 521

1 Monat lang 69,— DM für die Leerplatine mit Aufbauanleitung

512 KByte statische Ram/Epromkarte
- extern steckbar
- durchgeschliffener Bus
- statisch Ram/Eprom bestückbar
- in 64 K Schritten aufrüstbar
- adressbereich frei wählbar

Leerplatine	DM 69,—
Bausatz	DM 178,—
Fertigkarte OK	DM 248,—
Rams 32K * 8	DM 25,—
Eproms 32K * 8	DM 15,—

Multi I/O
- 72 frei programmierbare I/O Leitungen
- batteriegepufferte Echtzeituhr
- Experimentierplatine
- adressbereich frei wählbar
- mehrere Karten einsetzbar

Leerplatine	DM 98,—
Bausatz Uhr	DM 169,—
Bausatz I/O	DM 149,—
Bausatz Uhr + I/O	DM 199,—
Fertigkarte Uhr	DM 199,—
Fertigkarte I/O	DM 179,—
Fertigkarte Uhr + I/O	DM 259,—

Zweitlaufwerk für alle Amigas
- Bausatz mit allen benötigten Bauteilen, Laufwerk und Gehäuse

Fertiglaufwerk 1035	DM 369,—
Fertiglaufwerk 1036	DM 398,—
Bausatz mit 1035	DM 299,—
Bausatz mit 1036	DM 325,—

Sonstiges	
Amiga 2000	DM 2498,—
NEC Multisync	DM 1698,—
Kabel Amiga-NEC	DM 18,95
Commodore 1081	DM 748,—
Golem Box	DM 995,—
PC Karte	DM 1248,—
Harddisk 20 MB incl. Controller	DM 850,—
Business Karte 20 MB	DM 850,—
Stecker 86 polig	DM 18.50
Druckerkabel	DM 16,70

Software	
DPaint II	DM 248,—
CLI-Mate	DM 76,—
The Pawn	DM 66,—
Superbase	DM 228,—
Shell	DM 118,—
Profimat	DM 99,—
Textomat	DM 99,—
Datamat	DM 99,—
Beckertext	DM 198,—

Epromer bald lieferbar (bis 27011)

Die Sonderaktion geht bis zum 14.10.1987

Wiederverkäufer wenden sich an:

Dr. Nix GmbH
COLONIX
Robert-Perthel-Straße 2 · 5000 Köln 60
Telefon (02 21) 17 20 13
Telex 8 886 674 coln d
Teletex 2 214 019 nix d

Unverb. Preisempfehlung
Händleranfragen willkommen
Weitere Produkte in der Vorbereitung

tröps Computer Technik

Ralf Tröps · Computertechnik · Pingsdorferstr. 141 · 5040 Brühl · Tel. 0 22 32 / 1 30 63 und 4 71 05

Timesaver Macro-Clock für AMIGA 1000



Timesaver Macro-Clock, was verbirgt sich hinter diesem ominösen Titel? 'Ach so, schon wieder eine Uhr', werden vielleicht einige sagen und weiterblättern, aber die Anwender, die Timesaver noch nicht kennen, sollten doch weiterlesen!

Ich möchte hier kurz die Leistungsfähigkeit von Timesaver umreissen. Ein Handbuch kann dieser Hardware-Test nicht ersetzen, dieses wird nämlich mit 70seitigem Umfang mitgeliefert.

Beginnen wir also mit der Beschreibung der Installation von Timesaver. Keine Angst, hierbei muß man weder löten noch schrauben – lediglich die Kraft zweier gesunder Daumen wird vorausgesetzt.

Entfernt man Timesaver aus der Verpackung, ist man vielleicht ein wenig enttäuscht. Es zeigt sich nämlich nur ein würfelförmiges Metallgehäuse von ca. 7x3x4 Zentimetern Kantenlänge. Also nichts besonderes! Aber Timesaver soll ja nicht etwas fürs Auge sein, sondern soll lediglich funktionieren. Außerdem verschwindet es nach der Installation sowieso aus dem Blickfeld.

Timesaver wird an der Unterseite des Amiga angebracht. Man entfernt also Monitor und alle andere störenden Gegenstände von der Oberseite des Rechners und dreht diesen um 180 Grad, daß er auf dem Kopf liegt. Nun nimmt man das Uhr-Modul und legt es auf die Gehäuseaussparung hinter dem Netzanschluß. Jetzt kommt die oben erwähnte Kraft der zwei Daumen zum Einsatz – Timesaver wird einfach in diese Aussparung gedrückt – und schon sind fünfzig Prozent der 'Einbauarbeiten' beendet.

Jetzt wird die Uhr nur noch mit dem Rechner verbunden.

Das geschieht mit dem mitgelieferten Kabel, welches anstelle des Keyboard-Kabels in den Rechner gesteckt wird. Das Verbindungskabel zur Tastatur wird am Timesaver in den bezeichneten Eingang 'gestöpselt', und fertig ist der Umbau.

Nachdem der Monitor wieder auf seinem angestammten Platz steht, kann nam die Möglichkeiten von Timesaver testen.

Der Amiga wird ganz normal mit Kickstart und Workbench gebootet. Um die Funktionen von Timesaver zu nutzen, muß man ins CLI. Hat man nun den CLI-Prompt auf dem Bildschirm, drückt man die 'Help'-Taste in Verbindung mit 'Fl0' und bekommt folgende Ausgabe:

```
;f1 = auto break
;f2 = macros
;f3 = auto macros
;f4 = auto date
;f5 = history store
;f6 = echo
;f7 = auto password
;f8 = timesaver
;press any key to toggle
;f1 f2 f3 f4 f5 f6 f7 f8
;on on off on on on off
```

Mit der Tastenkombination [HELP + F10] erhält man also eine Übersicht, der verfügbaren Funktionen inklusive einer Statuszeile, die Auskunft über die aktiven Macros gibt. Da wir gerade bei einer der hervorragenden Eigenschaften von Timesaver, den Macros sind, möchte ich kurz darauf eingehen.

Timesaver besitzt einen Ram-Bereich von 7 Kbyte, der dem Benutzer zur freien Verfügung steht. Es besteht die Möglichkeit, diesen Speicher mit selbst definierten Macros vollzuschreiben. Fünfhundert dieser Funktionen, bedingt durch die Mehrfachbelegung der Amiga-Tastatur, können ohne Probleme aktiviert werden, wobei die Erstellung eines Macros bei Timesaver wirklich keine Schwierigkeiten bereitet. Ein Macros kann minimal die Ausgabe eines Zeichens bewirken, oder zum Beispiel auch eine Startup-Sequence von mehreren Bildschirmseiten Umfang aufrufen.

Aber nun weiter in der Besprechung der oben aufgeführten Funktionen. An erster Stelle steht dort 'auto break' [F1]. Dieser auto break ist vergleichbar mit dem Unterbre-

chen des Bootvorganges mit der Tastenkombination [CTRL + D]. Ist diese Option ermöglicht, unterbricht der Amiga den normalen Startvorgang, führt erst sämtliche im 8 Kbyte grossen EPROM gespeicherten Funktionen sowie Auto Macros aus, und setzt erst danach die normale Start-Sequence fort.

Auto Macros [F3] sind vom Anwender definierte Macros, sofern diese Option aktiviert ist, beim Booten automatisch ausgeführt werden.

Auto Date [F4], sofern aktiv, schreibt beim Booten Datum und Uhrzeit auf den Bildschirm.

History Store [F5] speichert jede Tastatureingabe im RAM ab. Man hat durch diese Funktion die Möglichkeit, vorher eingegebene Befehle durch Drücken der Help-Taste in Verbindung mit [Cursor UP], diese wieder auf den Bildschirm zu bringen, zu editieren, oder einfach durch drücken von RETURN neu zu starten. Ein Zeileneditor, wie ihn das SHELL von Metacombo (siehe Test in Kickstart Nr. 8) beinhaltet, wird bei dieser Uhr gratis mitgeliefert.

Echo [F6] dient lediglich dazu, Bildschirmausgaben, die Funktionen von Timesaver betreffen, zu unterdrücken.

Auto Password [F7] mag für die Anwender von Interesse sein, die verhindern wollen, das Hausfrau, Hund oder Kleinkind unbefugt Zugang zum Rechner erhalten. Ist diese Option aktiv, unterbricht der Amiga das Booten und wartet so lange, bis das richtige Passwort eingegeben wird. Danach wird die normale Start-Sequence ausgeführt.

Der letzte Punkt 'Timesaver' [F8] dient dazu, das Modul komplett abzuschalten. Es kann durchaus vorkommen, daß Tastenbelegungen von Anwender-Programmen, zum Beispiel Textverarbeitung, identisch sind mit Aktionstasten von Timesaver-Macros. Der Rechner kann dann nicht entscheiden, ob ein Macro ausgeführt werden soll, oder ob sich die gewählte Kombination von Tasten auf das laufende Programm bezieht. Für solche Situationen ist dann die Möglichkeit gegeben, Timesaver einfach per Tastendruck abzuschalten.

Zum Schluß möchte ich noch kurz auf die Hardware eingehen. Das Innenleben von Timesaver besteht aus einem 8-bit Microprozessor, der alle Operationen kontrolliert, einem 8 K EPROM, der das Programm beinhaltet (z. Zt sind erst 4 K davon belegt, die der Ausführung von fest gespeicherten Macros dienen) und einem 8 K großen RAM, der für die Aufnahme von selbst definierten Macros bestimmt ist.

Nicht zu vergessen ist natürlich der Low-Power Clock Chip zur Zeitausgabe, und ein spezieller Chip, der die Stromaufnahme für Timesaver kontrolliert. Dieser Chip sorgt dafür, daß bei Rechnerbetrieb Timesaver den Strom vom Amiga erhält, während er sonst auf Battariebetrieb umschaltet.

Beim Einschalten des Rechners, stehen also immer drei aktuelle Zeit, alle Macros sowie Funktionstastenbelegungen zur Verfügung.

Meiner Ansicht nach ist Timesaver ein sinnvolles und sehr nützliches Zubehör, das keine Joystick-Ports oder wertvollen Speicherplatz belegt und das mit ca. 180,– DM erschwinglich ist.

Urteil eines Anwenders: EMPFEHLENSWERT!

(F. Riedel)

Bezugsquelle:
DTM, Wiesbaden
IM 069/7071102

Leserecke &



Leserbriefe

Um eine bessere Schnittstelle zwischen der Redaktion und unseren Lesern zu schaffen, haben wir diese Seiten ins Leben gerufen. Die Leserecke soll Ihnen für Veröffentlichungen oder spezielle Fragen zur Verfügung stehen. Sie können uns auch Anregungen oder Kritikpunkte mitteilen.

Aus Zeitgründen können wir leider nicht alle Fragen beantworten, deshalb geben wir diese an unsere Leser weiter. Wenn Sie die Antwort(en) wissen, wären wir Ihnen dankbar, wenn Sie uns diese schriftlich übermitteln könnten. Wir werden die Antworten bereitwillig an unserer Leser weitergeben. Ebenso können Sie sich mit dem Fragesteller direkt in Verbindung setzen, da dieser meistens auf die Lösung seines Problems wartet. Scheuen Sie sich nicht, auch noch so scheinbar unsinnige Fragen uns zuzusenden, andere Leser werden Ihnen dankbar sein.

Die Redaktion

Ich habe folgendes Problem: Mir wurde ein 5,25-Zoll-Laufwerk (ohne alles, nur die Mechanik) geschenkt, welches ursprünglich für den Einbau in den Ericson PC gedacht war, wo jetzt ein Streamer sitzt, was immer das auch sein mag. Ich möchte dieses Laufwerk für meine Amiga 1000 benutzen, um damit Software wie dBaseIII, Wordstar, Frame Work usw. zu nutzen. Was muß ich tun, um dieses Laufwerk wo und wie an meinen Computer anschließen zu können? Zumal der Diskdrive-Anschluß schon von einem zweiten 3,5-Zoll-Laufwerk belegt ist. Wie mache ich dem Computer klar, daß dies ein 5,25-Zoll-Laufwerk ist, und er doch bitte die darin eingelegte Diskette im MS-DOS-Format lesen soll? Erledigt diese Aufgaben ein IBM-Emulator?

Thorsten Teigeler, Groß-Zimmern

Sehr geehrter Herr Teigeler, zur Beantwortung Ihrer Fragen, die sicherlich mehrere Leser interessiert: Um es vorweg zu sagen, das Laufwerk kann nicht ohne größeren Aufwand an den Amiga angeschlossen werden. Das liegt daran, daß die Amiga-Laufwerke eine Zusatzplatine besitzen, die erweiterte Steuermöglichkeiten erlaubt. Das Betriebssystem des Amiga benötigt die Zusatzschaltung zum getrennten Ansteuern der Laufwerksmotoren und zum Erkennen eines Diskettenwechsels. Die Möglichkeit zum Nachbau dieser Platine besteht durchaus und ist eine gar nicht so komplizierte Angelegenheit. Der Anschluß des modifizierten Laufwerks kann aber in Ihrem Falle nur erfolgen, wenn das externe Laufwerk einen durchgeführten Bus besitzt. Ansonsten kann das 5,25-Zoll-Laufwerk gegen das mit 3,5 Zoll ausgetauscht werden. Zwischen beiden Laufwerken besteht im Grunde genommen keinerlei Unterschied, vorausgesetzt, das 5,25-Zoll-Laufwerk ist in der Lage, 80 Tracks zu verarbeiten. Ein 5,25-Zoll-Laufwerk arbeitet mit den IBM-Emulator, so daß PC-Software geladen werden kann.

Die Redaktion

Ich besitze noch einen RGB-Farbmonitor von Cabel, den ich für den Amiga 2000 nutzen möchte. Auf der Rückseite des Monitors ist ein Video-Din Stecker. Meine Frage an Sie ist, was mache ich mit dem Pal-Sync-Signal (Pin des Din Steckers), welchem Signal beim Amiga entspricht dieses Signal?

Jakob Berger, 5800 Hagen

Herr Berger, das Pal-Sync-Signal entspricht dem Composite-Sync-Signal beim Amiga 2000. Mit diesem Wissen dürfte es kein Problem mehr sein, den Monitor am Amiga zu betreiben.

Christian Keller

Die nachstehenden Leser haben verschiedene Fragen und Anregungen, die Sie vielleicht beantworten können.

Martin Kinnen, Sepp-Herberger-Str. 27, 4018 Langenfeld

Kann ich den Tongenerator des Amiga von 8-Bit auf 16-Bit „aufrüsten"? Welche Möglichkeiten habe ich, einen Amiga an einen Synthesizer anzuschließen und dann eigene Samples einzuspielen? Wie kann ich einen gute Tonqualität erreichen, vom Preis einmal abgesehen?

Sehr geehrte Damen und Herren,
im Physikalischen Institut der Universität Bonn wird im Rahmen eines Forschungsprojekts der Elementarteilchenphysik ein neues Experiment aufgebaut. Die Größe dieses Teilchendetektors erfordert ein Leistungsfähiges Datenerfassungssystem, das wir mit Hilfe einiger 68000-CPU Karten (ca. 15) und dem VME-Bus aufgebaut haben. Als Steuerkonsole sind mehrere AMIGA 1000 Computer vorgesehen.

Wir stehen nun vor dem Problem, den AMIGA an den VME-Bus anzuschließen. Dabei ist die VME-Bus-Seite bereits vorhanden, aber für den AMIGA-Bus fehlt noch einige Information, die wir leider auch nicht in den 4 AMIGA Reference Manuals gefunden haben. So hoffen wir nun, daß Sie uns ein wenig weiterhelfen können:

- verfügen Sie über Informationen bezüglich der AUTOKONFIGURATION Möglichkeit?
- wie kann man über den externen Bus Interrupts erzeugen? (ohne in Konflikte mit den Koprozessoren zu geraten)
- verkraftet es der AMIGA, wenn über den Bus ein BERR-Signal (Bus Error) kommt?

K. Honscheid

Betr.: Anfrage auf den Sound Sampling Text (Grundlagen)

Sehr geehrte Damen und Herren

Ich besitze seit einiger Zeit einen A-2000 und auch einen CD'Player. Meine Frage läuft nun darauf hinaus, daß ich mir keinen Sound Sampler kaufen will, sondern die digitalen Daten direkt im CD Player abgreifen will. Der CD Player ist ein Schneider CDP 7500.

Gibt es eine Möglichkeit und wenn geht es mit einem im Handel erwerblichen dazu passenden Softwareprodukt.

Ich glaube, daß es eine Frage ist, welche bestimmt auch einige andere Leser interesiert.

Michael Schiffer

Betr.: Anfrage nach Programmierhilfe und Hardware!

Ich sitze jetzt schon den ganzen Tag an meinem Amiga 2000 und es passiert als nicht das was passieren soll. Schon etliche Gurus, jetzt reicht's!!!!!! Ich brauche Eure Hilfe. Ich gehöre zu den Verrückten die versuchen Basicprogramme auf dem Amiga zu schreiben um ihn besser kennen zu lernen. Mein Problem ist nun folgendes, ich will, wenn ich den Computer anmache und eine Diskette einlege ein Basiccodeprogramm aufrufen, welches dann bei Eingabe des richtigen Codewortes ein Maschinenspracheprogramm nachladen soll. Dazu muß ich eine bestimmte Befehlsfolge in die Startup-Sequence reinschreiben, doch eben diese Befehlsfolge kenne ich nicht.

Ein anderes Problem ist, die Amiga Floppy.

Ich habe den Amiga 2000 mit einer internen und einer externen Floppy, eben diese exteren Floppy erkennt der Amiga immer als df2 an. Viele Programme arbeiten aber nur mit df0 und df1!

Ich hoffe auch das meine Probleme einigermaßen erklärt sind und bitte nochmals um Hilfe, da mir noch nicht einmal in Fachgeschäften Auskunft gegeben worden konnte, seid Ihr meine letzte Rettung.

Michael Schiffer

Ich bin (noch !) Anfänger auf dem AMIGA 500 und wende mich mit zwei ganz speziellen Fragen, die das AmigaBasic (bzw. die Grafikansteuerung unter AmigaBasic) betreffen, an Sie.

Also, komme ich zur ersten Frage: Wie kann ich (unter AmigaBasic !) den Halfbrite-Modus (mit 64 Farben) und/oder den Interlace-Modus (mit 400/512 Zeilen) ein- und auch wieder ausschalten??

Meine zweite Frage betrifft den Mauszeiger: ich kann ihn zwar in den „Preferences“ verändern (was zugegeben auch sehr bequem zu bewerkstelligen ist) aber wie kann ich diesen Mauszeiger mitten im Programm von Basic aus modifizieren (als Beispiel könnte ich die Programme „Aegis Sonix“ oder „Deluxe Paint“ nennen: hier wird, je nachdem, welchen Modus man anwählt, der Mauszeiger geändert; oder werden hier vielleicht irgendwelche Tricks benutzt??).

Eine letzte Frage habe ich dann allerdings doch noch: ist es auf irgendeine Weise möglich, den „Hold and Modify“ (HAM)-Modus von AmigaBasic anzusprechen?? Wie geht dieses überhaupt??

Stefan Tandecki

Kleinanzeigen

BIETE HARDWARE

Verkaufe dt. AMIGA-1000 (7 Mon.) VHB 1300 DM, 512 KB. Tel. 0421/622958

Disks zw.2,4-2,6 DM:0431569216

BIETE SOFTWARE

Amiga Public-Domain, ausführl. Liste DM 5,– Ralf Dornenwerth, Buzallee 49, 8918 Dießen

Amiga-Software write to Holger Schneider, Elsa-Brandströmstr. 10 6500 Mainz-Gonsenheim

TAUSCH

Suchen, tauschen, (ver)kaufen Amiga Public Domain Info:10DM Beschr. der Prog. über 300 Screens = wird durch 1 Disk ersetzt. Private usergroup AIT/050418229 Ziegeleiweg 32, 3257 Springe 4

KONTAKTE

Suche Kontakt zu Amiga-Grafik-Anwendern im Raum F, HG, FB. Erfahrung-u.Softwaretausch, evtl. Zusammenarbeit bei Spez.-Aufg. Tel. ab 18[30], 06081/8590

Inserentenverzeichnis:

Einkaufsführer

Hier finden Sie Ihren Commodore/Amiga Fachhändler

Anzeigenschluß Heft 11/87: 25.September 1987

1000 Berlin

RUNOW
Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/641 50 91

Gerhard u. Bernd Waller GbR
Computer & Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07 + 570 52 75

GMA mbH
040/2512416/17
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

2160 Stade

BERGAU
Büromaschinen · EDV-Systeme
Neue Straße 5, 2160 Stade
Telefon: (04141) 23 64 + 23 84

2390 Flensburg

ecl electronic computer laden ohg
Norderstr. 94-96 · D-2390 Flensburg
(04 61) 2 81 81 & 2 81 93

2800 Bremen

Berthge & Strutz KG
St.-Jürgen-Straße 46 – 50
2800 Bremen 1
Tel. 04 21 - 70 00 57 - 59

2940 Wilhelmshaven

Radio Tiemann
Commodore-Systemfachhändler
Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

2960 Aurich

FRANZ-JOSEF KIRFEL
GEORGSWALL 18 a
2960 AURICH 1
TELEFON 0 49 41 - 6 23 55

3000 Hannover

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36

3250 Hameln

Witte Bürotechnik
Inh. Günther Kessels
Deisterstr. 53
3250 Hameln 1
Tel. 0 51 51 - 1 20 22 / 1 20 23

3470 Höxter

Schidlack + Sohn KG
Am Markt 8 · 3470 Höxter
Telefon 0 52 71 - 79 29 / 29 29

3500 Kassel

Hermann Fischer GmbH
Commodore-Systemfachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00



4000 Düsseldorf

CCS Commodore
Computer Schule GmbH
Immermannstr. 65
4000 Düsseldorf 1
Tel. 02 11 - 35 32 15

Hier könnte
Ihre Anzeige
erscheinen
Anruf genügt: Heim - Verlag
Tel. 0 61 51 - 5 60 57

4200 Oberhausen

LaSch
das Buch und Software Haus
Inh. Rainer Langner u. Franz Schnitzler GbR
Nohlstraße 76 · 4200 Oberhausen 1
Telefon 02 08 / 80 90 14

4600 Dortmund

Elektronik
Computer
Fachliteratur
AMIGA-Software
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84
city-elektronik

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4800 Bielefeld

hardware software organisation service **CSF**

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106–108
4800 Bielefeld 1
Tel. (0521) 61663

5000 Köln

BÜROMASCHINEN **braun**
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 0221/219171

5060 Bergisch-Gladbach

Computer Center
Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 02202-35053

5200 Siegburg

Computer Center
Luisenstraße 26
5200 Siegburg
Telefon 02241/66854

5500 Trier

bürocenter **LEHR**
Güterstr. 82 - 5500 Trier
☎ 0651-25044
Fordern Sie unsere Zubehör-Liste an.

5768 Sundern

C.S.C.
Computer & Software-Center
Hauptstr. 2 · 5768 Sundern
Telefon 02933-2046

6000 Frankfurt



GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Filiale Frankfurt	Filiale Hanau
Hartmann-Ibach-Str. 63	Steinheimer Str. 22
6000 Frankfurt 60	6450 Hanau
Tel.: (069) 462041	Tel.: (06181) 24826

Hardware Software Service **Computer bco**
Büro-Computer + Organisations GmbH
Commodore TOSHIBA ATARI OKIDATA
Ihr Partner, wenn es um Computer geht
6000 FRANKFURT/M. 1, Oeder Weg 7-9, ☎ 069/55045657

6200 Wiesbaden

Henneveld KG

Schossbergstr. 21
6200 Wiesbaden
Tel. 06121-2770

Poststraße 25
6200 Wiesbaden-Bierstadt
(06121) 560084
fax (06121) 563643



Werbung und EDV GmbH

AUTORISIERTER COMMODORE SYSTEM-HÄNDLER



Commodore

6250 Limburg

wir
bürosysteme vertriebs gmbh
diezer strasse 10
6250 limburg
tel. 06431-20030

6380 Bad-Homburg

PDC GmbH
Produkte u. Details Computerverbund
Louisenstr. 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Tel. 06172-24748
Bad-Homburgs erster Commodore Computerladen

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Autorisierter Commodore-Händler
Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon 06181-45293

6500 Mainz

Henneveld KG

Münsterstr. 15
6500 Mainz
Tel. 06131-2401

6520 Worms

Georg Steinmetz oHG
Neumarkt 4 + 10 · 6520 Worms
Telefon 06241-6868

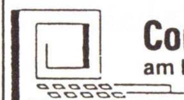
6680 Neunkirchen

Shop 64
Computer GmbH
Saarbrücken * Saarlouis
Homburg * St. Ingbert
Neunkirchen
06821/23713
Commodore Systemhändler

6750 Kaiserslautern



6800 Mannheim



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (0621) 850040 · Teletex 6211912

Computer-Center
am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim 1
Tel. (0621) 20983/84



Wo ist der Amiga-Fachhändler in meiner Umgebung?

Hier könnte Ihre Anzeige erscheinen
Rufen Sie uns an: Heim-Verlag 06151/56057

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.
SCHREIBER ELEKTRONIK
Alte Poststraße 2, 7000 Stuttgart-City
1. Stock, Telefon (0711) 22 70 99

7140 Ludwigsburg

B D T
BÜRO-DATEN-TECHNIK-VERTRIEBS GMBH
Kurfürstenstraße 18 · 7140 Ludwigsburg
Telefon 0 71 41 - 2 50 74

7500 Karlsruhe

MCT
Ges. f. Microcomputer
Troilingerstr. 3 · 7500 Karlsruhe 41
Tel. 07 21 / 47 27 95

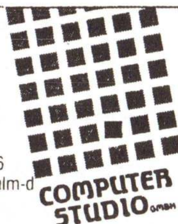
7700 Singen

Schellhammer GmbH
Freibühlstr. 21 + 23
7700 Singen
✆ 0 77 31 / 82 02 - 00

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm



EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstr. 28
7900 Ulm/Donau
Tel. 07 31 / 2 80 76
Telex 712 973 csulm-d

7920 Heidenheim

Das Büro
Kastorstr. 11
7920 Heidenheim/Brenz
✆ 0 73 21 - 4 40 15

7990 Friedrichshafen

KUMATRONIK Datentechnik
Schwabstr. 19
7990 Friedrichshafen
Telefon 0 75 41 - 7 20 41

8000 München

ludwig
COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

8120 Weilheim

COMPUTER STUDIO HUTTER GMBH
MÜNCHNER STR. 12 · 8120 WEILHEIM
TELEFON 08 81 / 12 23

8330 Eggenfelden

Hot Space
Computer-Centrum
R. Lanfermann
Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21 / 65 73
Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71 / 7 16 10
Innstraße 4
8341 Simbach
Telefon 0 85 71 / 44 10

8390 Passau

Zimmermann elektroland
8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

8400 Regensburg

Computer-Laden
Karl Steinmetz
Gewerbepark C 62 · 8400 Regensburg
Telefon 09 41 - 4 82 99



Wo ist der
Amiga-Fachhändler
in meiner Umgebung ?

Hier könnte Ihre Anzeige erscheinen
Rufen Sie uns an: Heim-Verlag 06151/56057

8400 Regensburg

Zimmermann elektroland
8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

8460 Schwandorf

Büro- u. Datensysteme GmbH
Ettmannsdorfer Str. 8
8460 Schwandorf
Tel. (0 94 31) 2 04 78 / 4 18 43

8500 Nürnberg

Orgaplus Datenverwaltung
G. Gailer KG
Fürther Str. 54 – 56
8500 Nürnberg 80
✆ 09 11 - 27 06 20

8600 Bamberg

BÜRO-ZENTRUM A+R KUTZ
Am Kranen Tel. 09 51 / 27 808
8600 Bamberg

8700 Würzburg

SCHOLL BÜROTEAM
Hardware · Software
Service · Schulung
computer center
am Dominikanerplatz
Ruf (09 31) 5 04 88

Schweiz

CH-8021 Zürich

Senn Computer AG
Langstrasse 31
Postfach
CH-8021 Zürich
Tel. 01 / 241 73 73
Telex 814 193 seco

PUBLIC DOMAIN SERVICE

Für den AMIGA gibt es schon eine Unmenge von Public-Domain-Programmen; manche Anbieter haben über 100 Disketten in Ihrem Programm. Die verschiedenen Sammlungen sind jedoch zum Teil nicht sortiert und in sich sehr unübersichtlich. Die Disketten wurden meist in den USA zusammengestellt (z.B. die 'Fish-Disks'), andere stammen aus den unterschiedlichsten Quellen. Dies hat dazu geführt, daß sich einige Programme auf mehreren Disketten wiederfinden. Hinzu kommt eine Vielzahl von Updates, also verbesserten Programmversionen, die die alten überflüssig machen. Ein weiterer Kritikpunkt an den meisten bestehenden Sammlungen ist die fehlende oder unzureichende Dokumentation, mit denen die Programmsammlungen angeboten werden. In den Listen steht oft nur eine Sammlung von Namen, deren Bedeutung jedoch nur selten deutlich wird. Fast immer fehlen Angaben über die Programmiersprache, den verwendeten Compiler und die notwendige Kickstartversion.

Aus diesem Dilemma soll Ihnen der Public-Domain-Service der KICKSTART helfen. Die Disketten enthalten ausschließlich auf ihre Funktionstüchtigkeit getestete Programme. Die einzelnen Disketten werden nach festen Kriterien zusammengestellt. Jede Diskette hat also einen Schwerpunkt (z.B. Lehrgänge, Bilder-Show, C-Programme, Spiele, u. ä.). Außerdem werden Angaben über die Programmiersprache, den verwendeten Interpreter oder Compiler usw. gemacht.

Diese Aktion, die uns einige schlaflose Nächt gekostet hat, soll Ordnung in die bestehende Public-Domain-Software bringen, damit Sie den größtmöglichen Nutzen daraus ziehen können. Wir hoffen, daß Sie mit der Einteilung und dem Ergebnis zufrieden sind und würden uns über Ihre Anregungen zu diesem Thema sehr freuen. Schreiben Sie uns, wenn Sie einen Verbesserungsvorschlag machen wollen! Wir haben dafür ein offenes Ohr und werden sicher einige dieser Vorschläge umsetzen.

Die Programme der Disketten 1–20 laufen auf allen AMIGA-Computern mit Kickstart/Workbench 1.2. Allerdings sollten 512k Speicher vorhanden sein. Sollten dennoch irgendwelche Einschränkungen gelten, werden wir dies bei den betreffenden Programmen angeben.

Das aktuelle Angebot

PUBLIC DOMAIN SERVICE

Diskette 1: C SOURCE

Eine Programmsammlung, die besonders dem Anfänger zeigt, wie man Intuition programmiert. Die Beispiele liegen als C-Quellcode und auch als fertige Programme vor, die sofort gestartet werden können.

Diskette 2: Spiele

- YachtC (Würfelspiel für 4 Personen)
- Puzzle
- Missile (verteidigen Sie Ihre Stadt, starker Sound)
- TriClops (sehr schönes 3D-Spiel)
- Breakout (3D-Effekt mit Brille)
- Trek73 (bekannte Star Trek Variante)

Diskette 3: Spiele

HACK: Das bekannte Textadventure, das ursprünglich auf UNIX-Rechnern erstellt wurde, liegt hier als spezielle Grafikversion für den Amiga vor.

Diskette 4: Terminal-Programme

KERMIT: Bekanntes, luxuriöses Terminalprogramm (drei verschiedenen Versionen, mit Source-Code in C).

Diskette 5: Terminal-Programme

- WOMBAT (VT102/52 Emulator, XModem, Autodial)
- VT100 (grafikfähig, Source in C)
- TermPlus (XModem, Source in C)
- DG210 (Data General D-210 Emulator)
- Ahost (XModem, Kermit)
- TeK4010 (XModem, VT100)

Diskette 6: Terminal-Programme

- Speech Term (spricht den empfangenen Text, XModem)
- StarTerm (mit Phone, Duplex, XModem)
- Argo Term
- PD Term (Source in C)
- AmigaDisplay
- Kermit

Diskette 7: UTILITIES

- QuickCopy (gutes Kopierprogramm)
- DirUtil (File-Copy)
- FileZap (File-Monitor)
- DiskZap (Disk-Monitor)
- DiskSalv (Diskettenretter)
- System-Monitor
- CSH (UNIX-ähnlicher Shell)

Diskette 8: Spiele

Monopoly: Sehr schöne Grafik, einfache Mausbedienung (Source in A-BasiC).

Diskette 9: Grafik-Show

- Grafik-Show mit bekannten Cartoons

Diskette 10: Grafik-Show

JUGGLER DEMO: Ein bewegliches Männchen jongliert mit drei verspiegelten Kugeln, sehr schöne Demo.

Diskette 11: Grafik-Show

RAY TRACERS: Wunderschöne räumliche Bilder, die auf einer VAX berechnet und auf den AMIGA übertragen wurden.

Diskette 12: Grafik

– Digitalisierte Bilder mit erstaunlicher Qualität (IFF-Format)

Diskette 13: Grafik

– Sehr schöne Bilder-Show (IFF-Format)

Diskette 14: EDITOR

Bekannter Texteditor MICROEMACS Version 30. Viele Features: Search/Replace/Copy.

Diskette 15: Grafik

verschiedene mit dem AEGIS-ANIMATOR erstellte Filme incl. PLAYER zum Abspielen der Filme (einige Filme benötigen auf einem AMIGA 1000 mehr als 512k Speicher!).

Diskette 16: Sprachen

XLISP 1.7 (neueste Version) mit ausführlicher Anleitung (über 50k)

Diskette 17: Sprachen

MODULA-2: Pre-Release eines Modula-Compilers mit verschiedenen kleineren Beispielprogrammen, die als Source-Code vorliegen.

Diskette 18: Grafik

MANDELBROT

Diskette 19: Grafik-Show

– Sehr schöne digitalisierte H.A.M.-Bilder

Diskette 20: Grafik-Show

'Fred the Baker und Rose's Flower Shop': COMIC-Film, der die Multitasking-Fähigkeiten des AMIGA erklärt.

Diskette 21: AMIGA-Tutor

Einführung in die Bedienung des AMIGA 500. Ein farbenfroher Lehrgang, der ganz am Anfang beginnt und mit vielen Bildern und Grafiken die Grundbegriffe des AMIGA erklärt (für Anfänger, komplett in Deutsch).

Diskette 22: Sprachen

MVP-FORTH und C-FORTH
C-Forth ist ein recht leistungsfähiger FORTH-Interpreter, der auch als Quelltext vorliegt.

Diskette 23: Grafik-Show

Viele abwechslungsreiche Motive in verschiedenen Auflösungen, verpackt in einer Grafik-Show (startet automatisch!)

Diskette 24: Grafik-Show

Sehr schöne digitalisierte Frauengesichter (startet automatisch!)

Diskette 25: UTILITIES

CLOCK, Portar, Checkmodem, PopCLI, MACView, Kickbench, Disassembler, Tracker und vieles mehr

Diskette 26/27: Grafik-Show

Auf zwei randvollen Disketten erleben Sie eine einmalige Dia-Show mit hervorragend digitalisierten futuristischen Bildern in voller PAL-Auflösung. Dazu gibt es stimmungsvolle, spärische Musik (startet automatisch!)

Diskette 28: Editoren

Auf dieser Diskette befinden sich einige schöne Editoren (UEDIT, MED, BLITZ) mit dazugehörigen Zeichensatz-Utilities (UeTurbo, Blitzfonts, Pearlfonts).

Diskette 29: UTILITIES

PrtDrvGen: erstellt Drucker-Treiber
DropShadow: jedes Fenster bekommt einen Schatten
MemClear: löscht den Speicher
ScreenSave: speichert den Bildschirm auf Diskette
Compress: komprimiert Programme

Diskette 30: Sound-Demos

Changing Minds, Joan Lui, Miami Vice II, Respectable, Holiday

Diskette 31: SOUND DEMO

Dieses Programm erzeugt naturgetreue Geräusche, die über die Tastatur, wie auf einem Klavier anspielen werden können.

Diskette 32: SOUND-DEMOS

Mit einer Demo-Version von SoundScape können digitalisierte Musikstücke abgespielt werden. Die Qualität ist wirklich erstaunlich.

Diskette 33: GRAFIK-SHOW

Einige sehr gute, mit Deluxe Video erstellte Filme. Der benötigte PLAYER ist auch auf der Diskette.

Diskette 34: SPIELE

Tunnel Vision (ABasiC) u.a.

Diskette 35: UTILITIES

AHA-Copy V1.0 (Nibble- u. Parameter-Copy, IBM-Format, Track 0—81)

Diskette 36: CAD

MCAD ist ein wirklich gut gemachtes CAD-Programm, daß jedoch nur im Interlace-Modus läuft. Es bietet die einfachen Zeichenfunktionen und Features wie Zoom, Group, Ungroup, Grid, Move, Rotate). Auf der Diskette sind mehrere Dokumente, die das Programm erklären.

Versandbedingungen:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

– Legen Sie pro bestellter Diskette DM 10,– als Scheck bei.
– Fügen Sie dem Betrag folgende Versandkosten (für Porto und Verpackung) bei: Inland DM 5,–, Ausland DM 10,–.
– Zu dem Kostenbeitrag müssen wir leider pro Scheck eine Scheckgebühr von DM 0,50 berechnen. Verwenden Sie deshalb nur einen Scheck, auf dem die Gesamtsumme steht (z. B. für zwei Disketten DM 25,50).
– Für ausländische Besteller sei erwähnt, daß Sie mit Euroschecks auch in DM bezahlen können.
– Legen Sie unbedingt einen Aufkleber mit Ihrer vollständigen Adresse bei.

KICKSTART-Redaktion
Industriestr. 26
6236 Eschborn
Tel. 06196/41245

Wichtig!

Die Software wird nicht auf 'Noname'-Disketten geliefert, sondern auf Qualitätsdisketten der Marke FUJI FILM, mit denen wir sehr gute Erfahrungen gemacht haben.

Impressum

KICKSTART

Chefredakteur:
Uwe Bärtels (Chefredakteur) (UB)
Markus Nerding (Stellvertreter) (MN)

Redaktion
Andreas Krämer (AK)
Gerald Carda (GC)
Dipl. Ing. Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)

Herausgeber:
„MERLIN" Computer GmbH
Industriestraße 26
Postfach 55 69
6236 Eschborn
Tel.: 0 61 96 / 48 18 11
FAX: 0 61 96 / 4 11 37

Ständige Mitarbeiter:
Christian Keller (CHK)
Christian Schormann (CS)
Wolf Dietrich (WD)
Roland Foerster (RF)

Public Relations:
Claus Peter Lippert

Verlag:
Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.: 0 61 51 / 5 60 57
FAX: 0 61 51 / 5 56 89

Verlagsleitung:
Hans-Jörg Heim

Anzeigenverkaufsleitung:
Uwe Heim

Anzeigenpreise:
nach Preisliste Nr. 1, gültig ab 1.7.86

Produktion:
Klaus Schultheis
Patricia Illing

Grafische Gestaltung:
Rosina Altkorn

Fotografie:
Klaus Ohlenschläger
Markus Nerding

Titelgestaltung:
Fabian & Maier

Titelfoto:
Rainer Spirandelli

Satz:
Mohr Pfungstadt

Druck:
Ferling Druck Darmstadt

Bezugsmöglichkeiten:
Zeitschriftenhandel, Kauf- und Warenhäuser, Commodore-Fachhändler oder direkt beim Verlag

Kickstart erscheint 11 mal im Jahr
Einzelpreis: DM 7,–, ÖS 56,–, SFr. 7,–
Das Jahresabonnement kostet DM 70,– inkl. Versandkosten + MwSt.
Europ. Ausland DM 90,– inkl. Versandkosten

VORSCHAU:

Software:

SCULPT 3D
ein RAY-TRACER Generator

Textverarbeitung mit PROWRITE

Hardware:

Ein Scanner für PC oder AMIGA

Die neue Festplatte von CLT im Test

Listings:

ANIMATOR
– Rotationskörper in C

Eine Shell für AZTEC C

VOCABLECRAFT, der Vokabellehrer (AmigaBasic)

Änderungen vorbehalten!

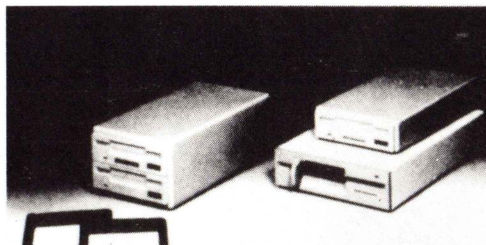
Ab 16. Oktober an Ihrem Kiosk!

Hattinger Str. 685
4630 Bochum 5
☎ 02 34/41 19 13
41 19 47

Soyka

DATEN TECHNIK

AMIGO	F1 3,5" Einzelfloppy	369,—
AMIGO	F2 3,5" Doppelfloppy	669,—
AMIGO	F5 5,25" Einzelfloppy	489,—
AMIGO	F1 Bausatz	289,—
AMIGO	F2 Bausatz	569,—
AMIGO	F1 Gehäuse	34,90
AMIGO	F2 Gehäuse	39,90
AMIGO	Sound (Mono)	129,—
AMIGO	Sound (Stereo Hifi)	199,—
AMIGO	Bootselector	39,90
AMIGO	Laufwerk NEC 1036a	259,—
AMIGO	F1 Bausatz mit Gehäuse	298,—



NEC 1036 A + Interface + Kabel + Stecker
+ Anleitung zum Selbstbau einer 3,5″

Amiga-Floppy 275,— DM

Amigo Laufwerke

* 1 Jahr Garantie
* Kunststoffgehäuse
* Amigafarbe
* Voll abgeschirmt
* abschaltbar
* Frontblende beige
* LED Steuerung wie internes LW
* „sehr zuverlässig" (Amiga Magazin)

Amiga 500	1148,—
Amiga 2000 mit Monitor	3190,—
Amiga 500 1 MB-Erweiterung	275,—
Amiga 500 1 MB-Erweiterung abschaltbar	289,—
Amiga 2000 1,5 MB-Erweiterung	198,—
Amiga Bücher (Reference Manuals):	
Intuition, Hardw. Exec	je 62,50
Libraries and Devices	88,—

PREISSCHLAGER

Deluxe Paint II 197,— Dig. View 2.0 297,—

Anwendung

AC-Basic	387,—
AC Fortran	597,—
Acquisition	597,—
Aegis Sonix 2.0	167,—
Amiga C	257,—
Amiga Dos Express	57,—
Analyse	198,—
Analyse Graph	297,—
Aztec C Dev. V3,4a	597,—
Aztec C Com. V3,4a	947,—
C-Monitor	197,—
City Desk	227,—
Cli-Mate 1,2	67,—
Devpac Assembler	197,—
Deluxe Music Con. Set	197,—
Deluxe Paint IIb	197,—
Deluxe Video V1.2	247,—
Digi Point	117,—
Digi View 2,0	297,—
Dynamic Cad	997,—
Dynamic Word*	377,—
Fast Fonts	97,—
Flip Side	97,—
Fortran	597,—
Galileo	207,—
Gizmoz	127,—
Grabbit	67,—
Instant Music	97,—
Lattice C V3,1	347,—
Lattice C prof.	887,—
Lisp	417,—
Logistix (Deutsch)	347,—
LPD Writer	247,—
Macro Assembler	197,—
Marauder II	77,—
Metacomco Pascal	227,—
Modula II Com.	597,—
Modula II Standart	177,—
Modula II Developers	297,—
MusicX*	547,—
Newio	477,—
Pagesetter Europ. Ver.	327,—
Planetarium*	97,—
Power Window	187,—
Printmaster Plus	107,—
Pro Video	347,—
Pro Write	247,—
Scribble	197,—
Seka	137,—
Shell (CLI deluxe)	137,—
Superbase (Deutsch)	227,—
Superbase Personal	297,—
Toolkit	97,—
True Basic	277,—
True Basic Dev. Toolkit	87,—
TV-Text	197,—
UBM Text V2,2	247,—
USCD Pascal	167,—
3D Videoscape	387,—
VIP Professional	447,—
Vizawrite (Deutsch)*	197,—
Zuma Fonts Vol. 1	67,—
Zuma Fonts Vol. 2	67,—
Zuma Fonts Vol. 3	67,—
WordPerfect 4.1	747,—

Spielesoftware

Adventure Constr. Set	67,—
Alien Fires	84,—
Alternate Reality „The city"	77,—
Archon II	77,—
Arena	77,—
Autoduell	77,—*
Bad Cat	57,—*
Barbarian	77,—
Bard's Tale	97,—
Beat It	27,—*
Black Cauldron	77,—*
Bureaucrazy	87,—
Challenger	27,—
Championship Baseball	87,—
Championship Basketball	87,—
Championship Football	87,—
Championship Golf	87,—
Chessmaster 2000	87,—
City Defence	27,—*
Cruncher Factory	27,—
Der Hauch des Todes	57,—*
Defender of the Crown	87,—
Deja Vu	87,—
Demolition	27,—
Dr. Fruit	27,—
Earl Weavers Baseball	97,—
Emerald Mine	27,—*
Faery Tale	97,—
Ferrari Formula One	87,—*
Fire Power	47,—*
Flugsimulator II (neue Vers.)	97,—*
Fortress Underground	27,—*
Galactic Invasion	47,—*
Garrison	67,—*
Gauntlet	87,—*
Golden Oldies	47,—
Goldrunner	87,—*
Grand Slam Tennis	87,—
Grand Prix	87,—*
Guild of Thieves	87,—
Gunship	77,—*
Hardball	87,—
Hollywood Poker	57,—
Indoor Sports	87,—*
Jet	87,—*
Kampfgruppe	107,—
Karate Kid II	67,—
Karate King	37,—
King of Chicago	97,—*
Land of Legends	97,—*
Leader Board	77,—
Little Computer People	57,—
Marble Madness	87,—
Mindbreaker	27,—
Mission Elevator	57,—*
Night Hawk	57,—
Pac Boy	27,—
Paperboy	87,—*
Phantasie	77,—
Phalanx	27,—
Planetarium	97,—*
Portal	87,—
Quintette	87,—
Quiwi	57,—
Quizam	67,—
Return to Atlantis	87,—*
Roadwar 2000	57,—
Rocket Attack	27,—
S D I	97,—
Shadow Gate	77,—*
Shanghai	77,—
Shooting Star	27,—
Sinbad and the thrown	97,—
Sky Fighter	57,—
Space Battle	27,—
Space Fight	27,—
Silicon Dreams	67,—*
Starglider	77,—
Strip Poker	77,—
Tenth Frame	77,—*
Terrorpods	67,—*
Testdrive	77,—*
The Final Trip	27,—*
The Surgeon	117,—
Turbo	47,—*
Typhoon	67,—
Uninvited	97,—
Winter Games	67,—
World Games	67,—

★★★ Harald Soyka ★★ Hattinger Straße 685 ★★ 4630 Bochum 5 ★★★